# DediProg SF Software User Manual

V 6.5

# Table of Content

**I. Introduction ..... 3**

**II. Software Installation Guide ..... 3**

2.1 Operating System Requirement ..... 3

2.2 USB Installation ..... 3

**III. DediProg SF Software Engineering GUI ..... 9**

3.1 Prepare the Environment ..... 9

3.2 Identify the Target SPI Flash ..... 10

3.3 Tool Bar Description ..... 11

3.4 Edit Window Description ..... 13

3.5 Configuration Window Description ..... 15

3.6 Supported Devices, Software Version, Firmware Version ..... 28

**IV. DediProg SF Software Production GUI ..... 29**

4.1 Search and Select ..... 31

4.2 Batch Config ..... 33

4.3 Single Site programming ..... 33

**V. DediProg Windows Command Line ..... 34**

5.1 Introduction ..... 34

5.2 How to Start ..... 37

5.4 Basic Switches ..... 37

5.5 Optional Switches ..... 38

5.6 Exit Code ..... 40

**VI. Specific Functions (SF600 and SF600*Plus*) ..... 41**

6.1Dual/Quad IO ..... 41

6.2 Hold Pin Status Setting ..... 41

**VII. Stand Alone Mode (SF600*Plus* only) ..... 42**

7.1 Project preparation ..... 42

7.2 Stand Alone programming ..... 46

**VIII. Firmware Support for Microsoft Windows ..... 46**

**IX. Revision History ..... 47**

**Important notice:**
This document is provided as a guide line and must not be disclosed without consent of DediProg. However, no responsibility is assumed for errors that might appear.

DediProg reserves the right to make any changes to the product and/or the specification at any time without notice. No part of this document may be copied or reproduced in any form or by any means without prior written consent of DediProg.

# I. Introduction

This user manual illustrates the usage of DediProg SF Software. The device connected when using this software can be used together with SF100, SF600, SF60*Plus* and Backup Boot Flash kit. Get more information for DediProg products and how to use them.
Please refer to our products specification, presentation and application notes on our website: www.DediProg.com

# II. Software Installation Guide

## 2.1 Operating System Requirement

Windows 8.1
Windows 8
Windows 7
Windows Server® 2008
Windows Vista®
Support both 32 bit and 64bit OS

## 2.2 USB Installation

2.2.1 Insert the installation CD or download the installation software from www.DediProg.com

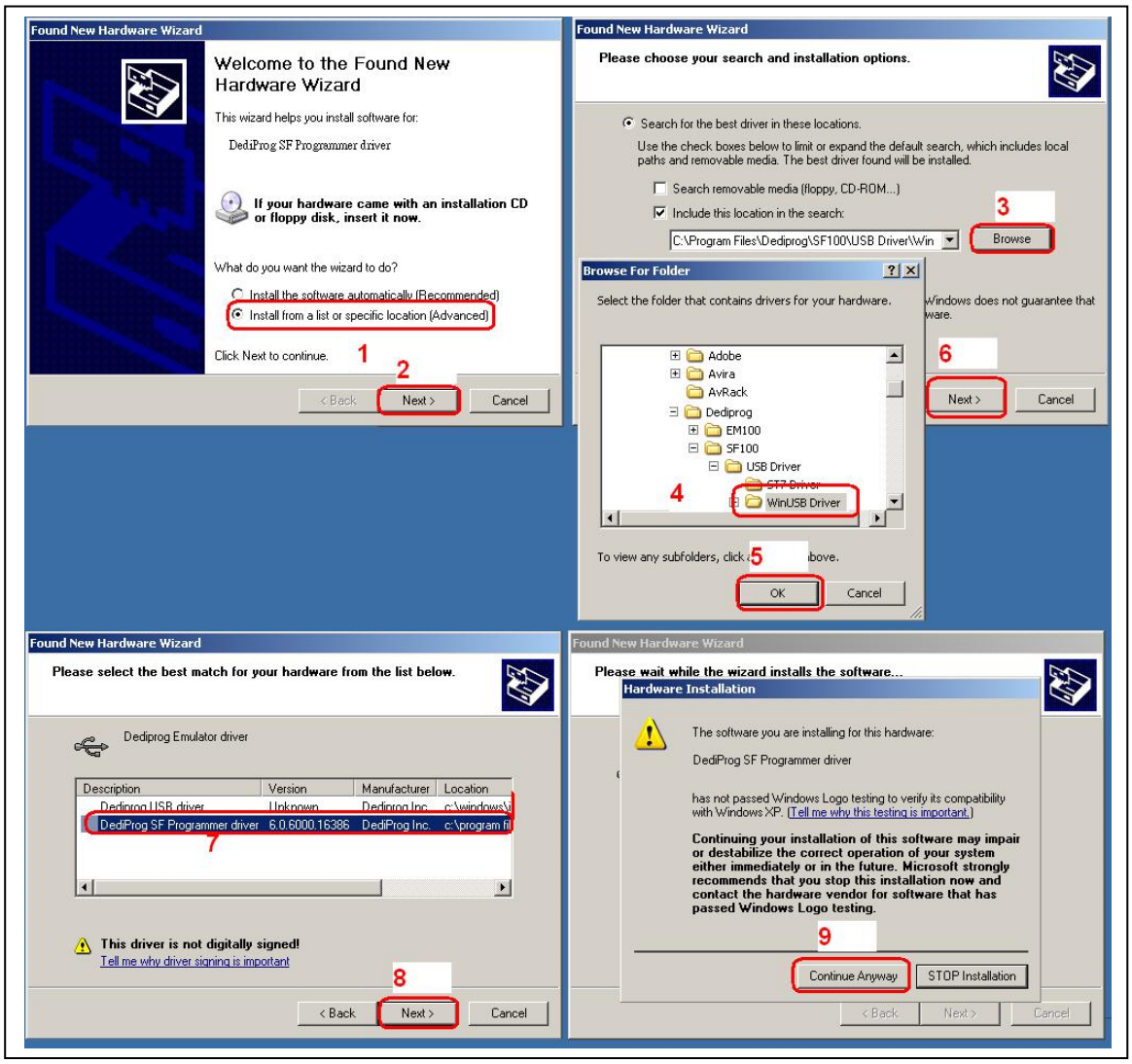
2.2.2 Execute SFx.x.x.msi file and click next until the installation is finished.

2.2.3 After step 2, plug in the USB cable to your computer and DediProg programmer, A Hardware wizard will show up as follow.

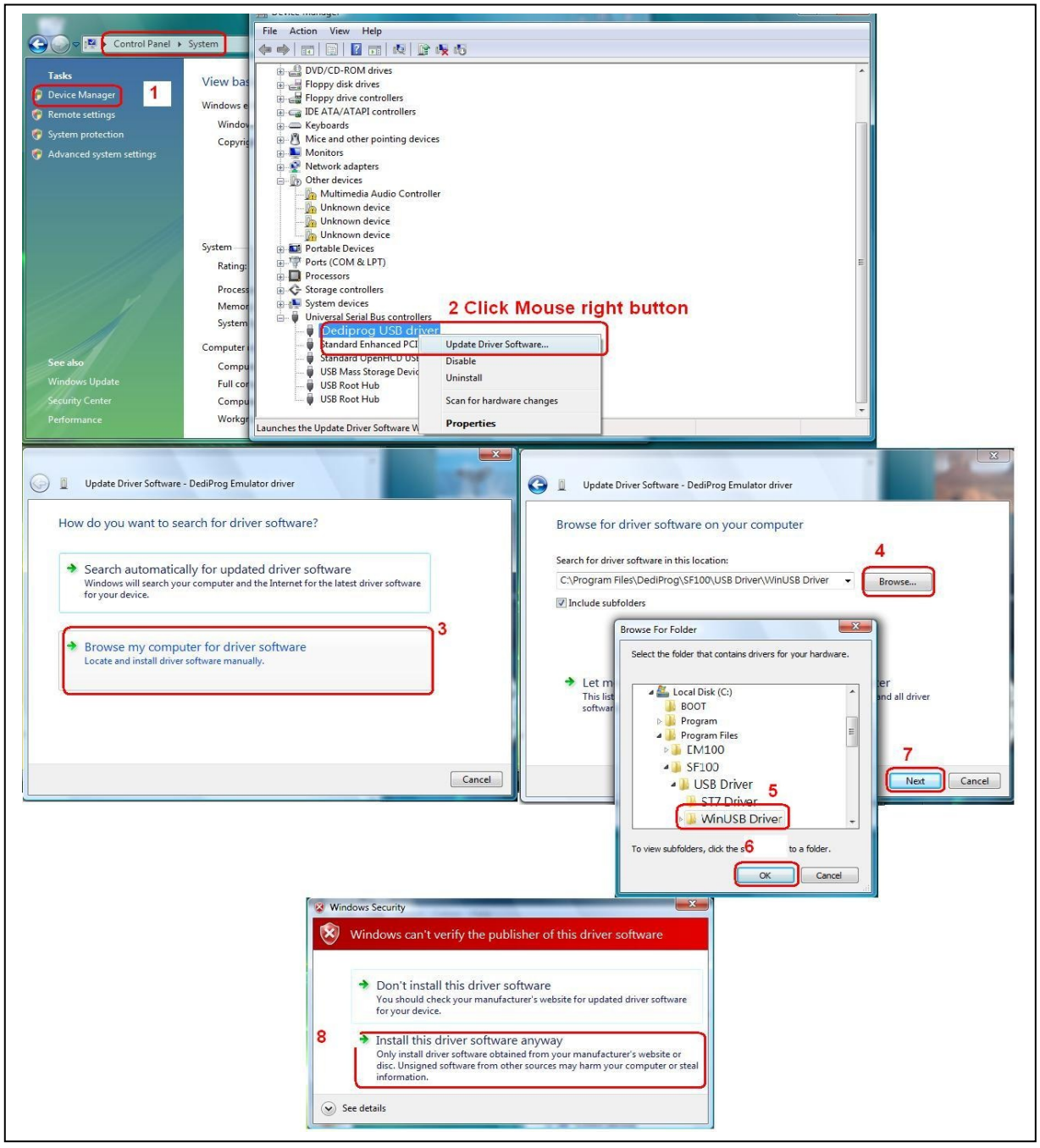
Found New Hardware Wizard
Welcome to the Found New Hardware Wizard
Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).
Read our privacy policy
Can Windows connect to Windows Update to search for software?
Yes, this time only
Yes, now and every time I connect a device
No, not this time
Click Next to continue.
< Back
Next >
Cancel

Follow the steps showed in the snap shots below to complete the installation.

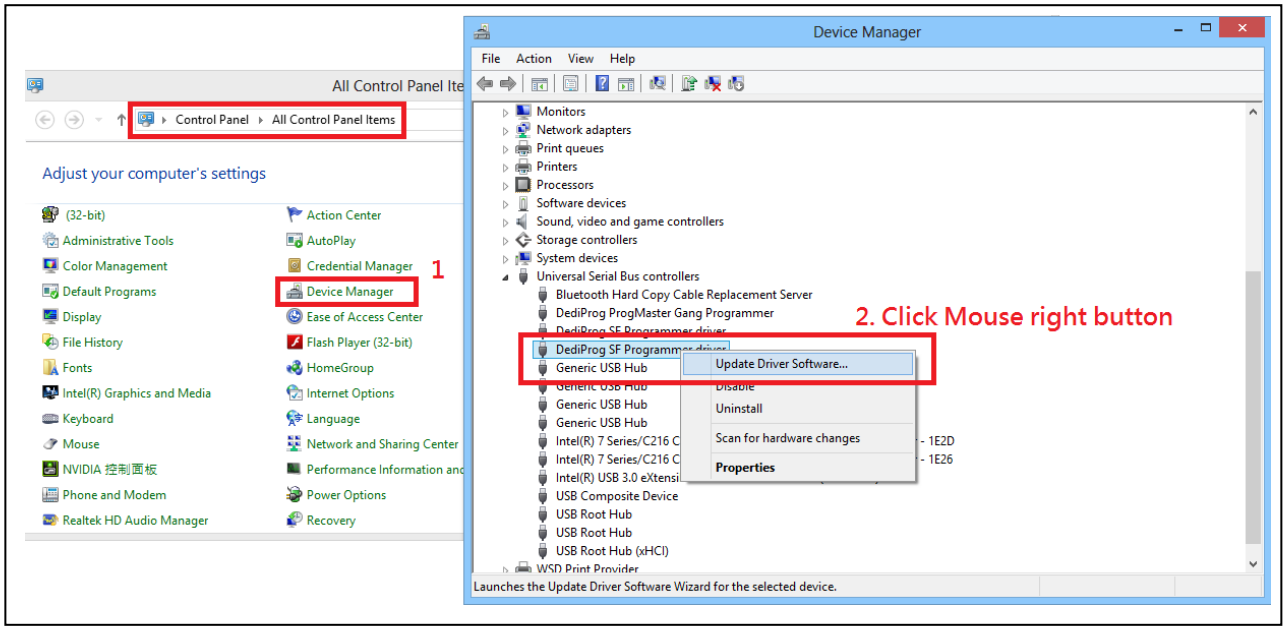
A. Win XP operation system:

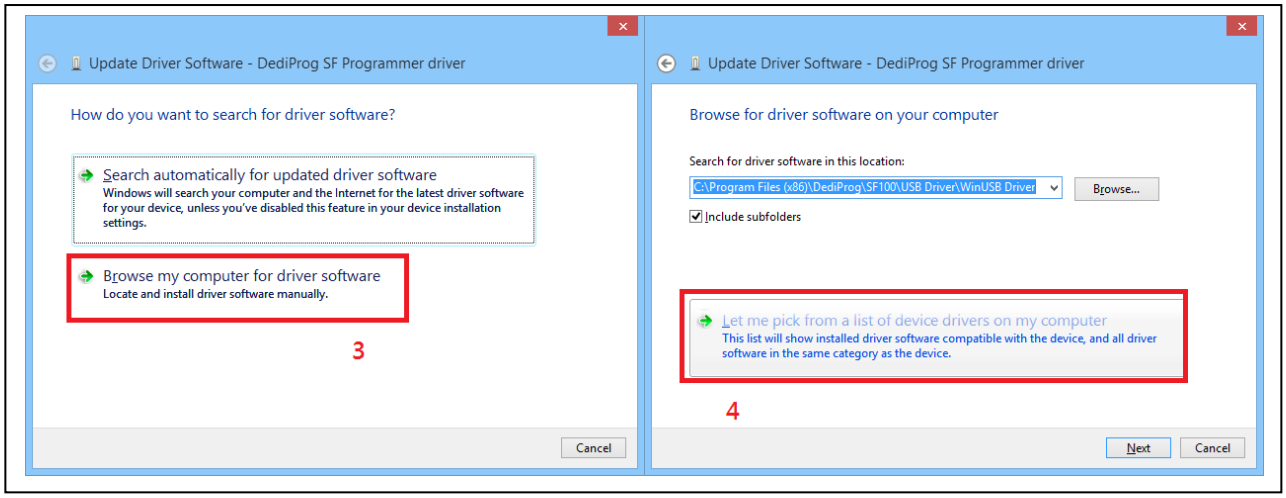

Found New Hardware Wizard
Welcome to the Found New Hardware Wizard
This wizard helps you install software for:
DediProg SF Programmer driver
If your hardware came with an installation CD or floppy disk, insert it now.
What do you want the wizard to do?
Install the software automatically (Recommended)
Install from a list or specific location (Advanced)
Click Next to continue.
1
2
< Back
Next >
Cancel
Found New Hardware Wizard
Please choose your search and installation options.
Search for the best driver in these locations.
Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.
Search removable media (floppy, CD-ROM...)
Include this location in the search:
C:\Program Files\Dediprog\SF100\USB Driver\Win
3
Browse
Browse For Folder
Select the folder that contains drivers for your hardware.
Adobe
Avira
AvRack
Dediprog
EM100
SF100
USB Driver
ST7 Driver
WinUSB Driver
4
To view any subfolders, click
5
above.
OK
Cancel
Windows does not guarantee that
ware.
6
Next >
Cancel
Found New Hardware Wizard
Please select the best match for your hardware from the list below.
Dediprog Emulator driver
Description
Version
Manufacturer
Location
Dediprog USB driver
Unknown
Dediprog Inc.
c:\windows\i
DediProg SF Programmer driver
6.0.6000.16386
DediProg Inc.
c:\program fi
7
This driver is not digitally signed!
Tell me why driver signing is important
8
< Back
Next >
Cancel
Found New Hardware Wizard
Please wait while the wizard installs the software...
Hardware Installation
The software you are installing for this hardware:
DediProg SF Programmer driver
has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)
Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.
9
Continue Anyway
STOP Installation
< Back
Next >
Cancel

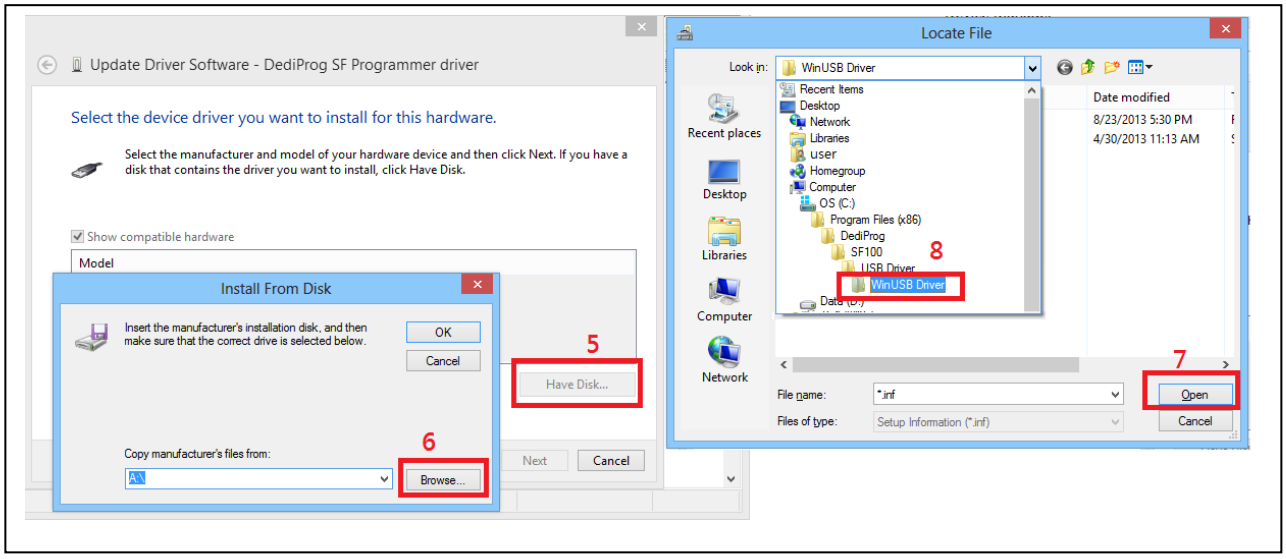
B. Win Vista & 7 Operation Systems:

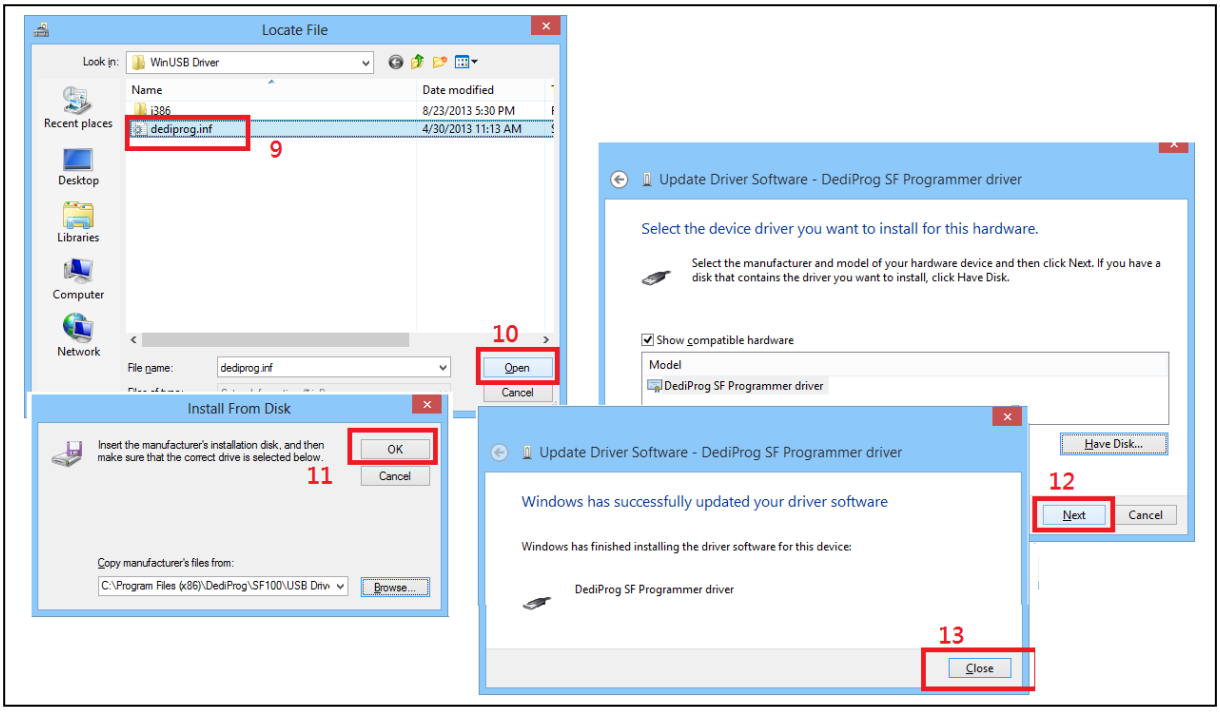


C. Win 8 & Win 8.1 Operation Systems:

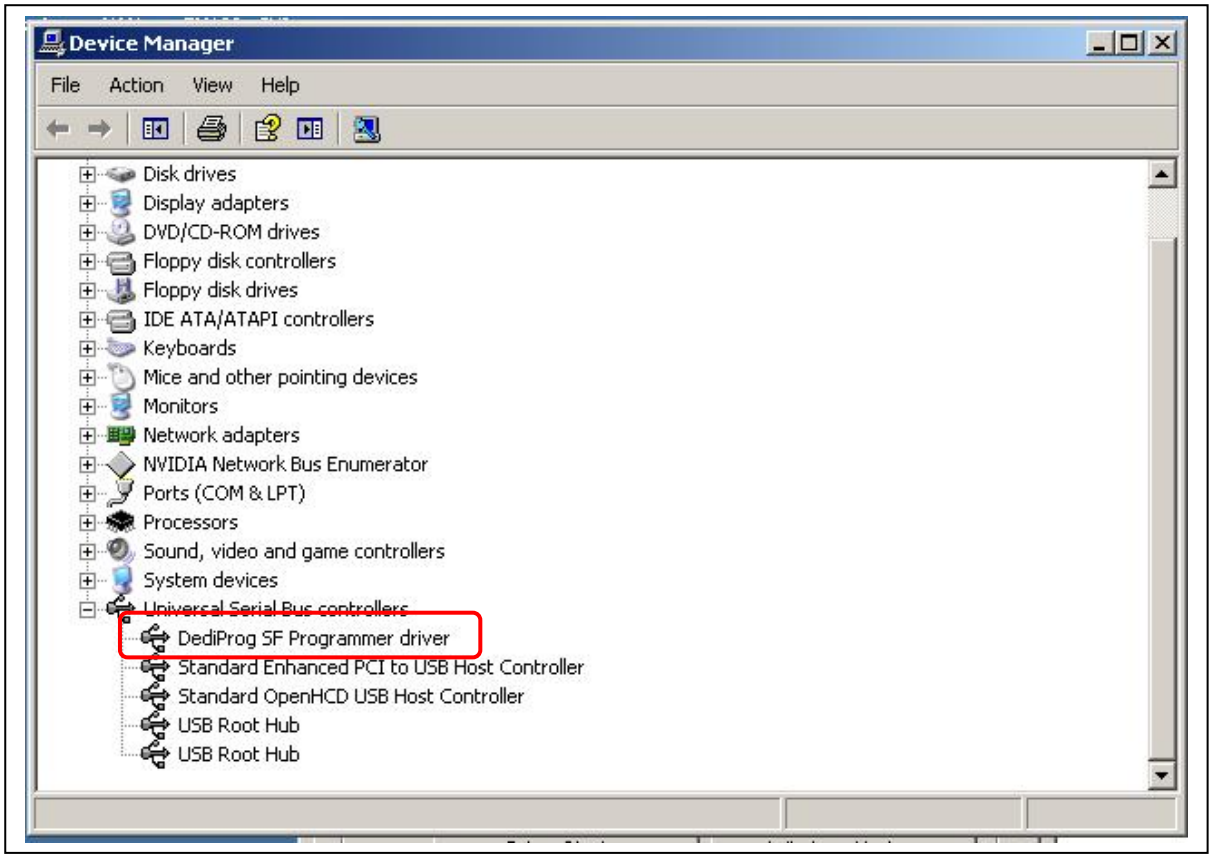




Update Driver Software - DediProg SF Programmer driver
How do you want to search for driver software?
Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.
Browse my computer for driver software
Locate and install driver software manually.
3
Cancel
Update Driver Software - DediProg SF Programmer driver
Browse for driver software on your computer
Search for driver software in this location:
C:\Program Files (x86)\DediProg\SF100\USB Driver\WinUSB Driver
Browse...
Include subfolders
Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all driver software in the same category as the device.
4
Next
Cancel

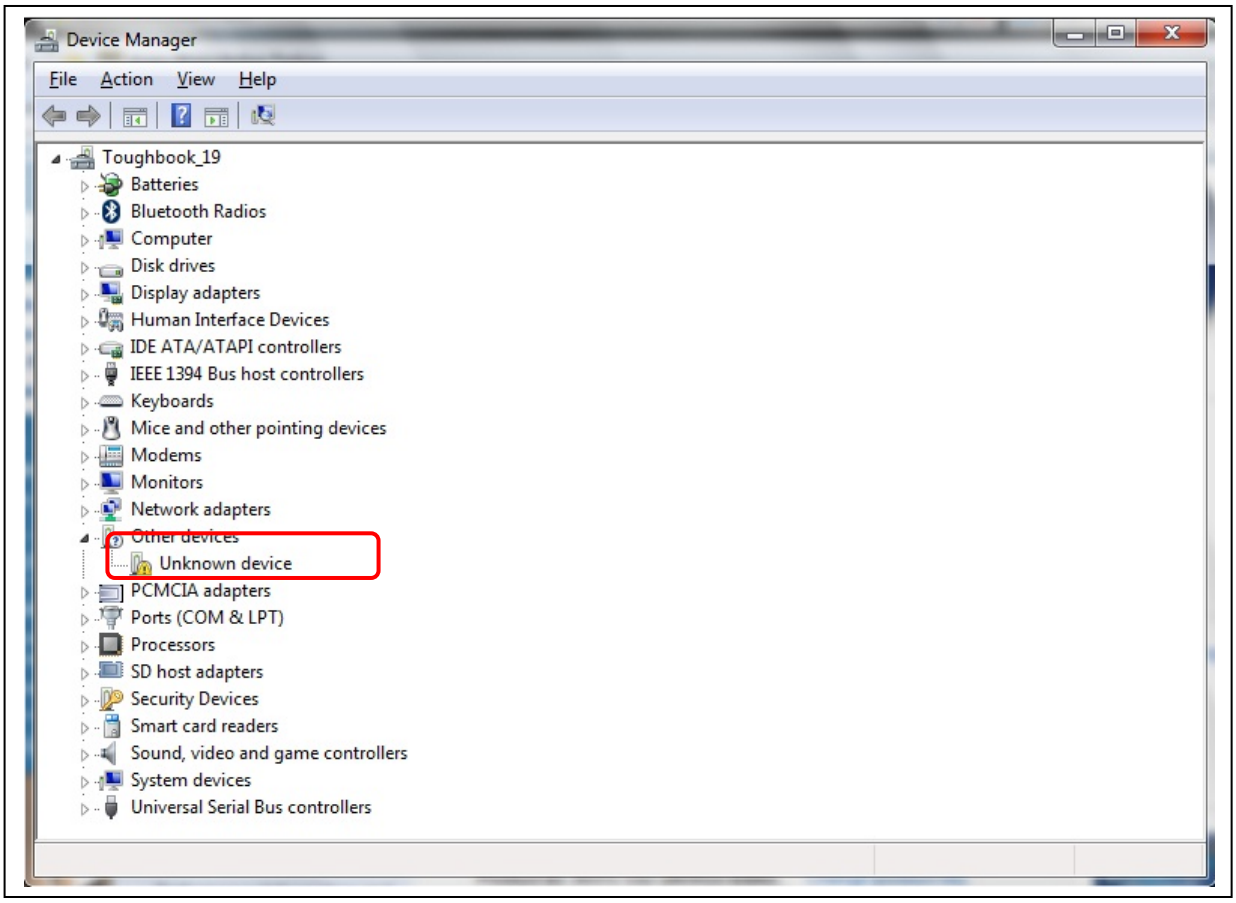





If the USB driver is installed properly, users can find "DediProg SF Programmer driver" under device manager when the programmer is plugged into the PC.



If no, please check “Other devices”. After selecting and installing, it will show up correctly under USB controllers.



# III. DediProg SF Software Engineering GUI

DediProg SF software is suited for SF100, SF600, SF600*Plus* and Backup Boot Flash Kit. The software only can be used to program serial flash memory as well as the downloading configuration contents to the reference SPI Flash embedded memory in SF600*Plus* for stand alone programming purpose. After the software and USB driver installed, please follow the steps as below before running the software.

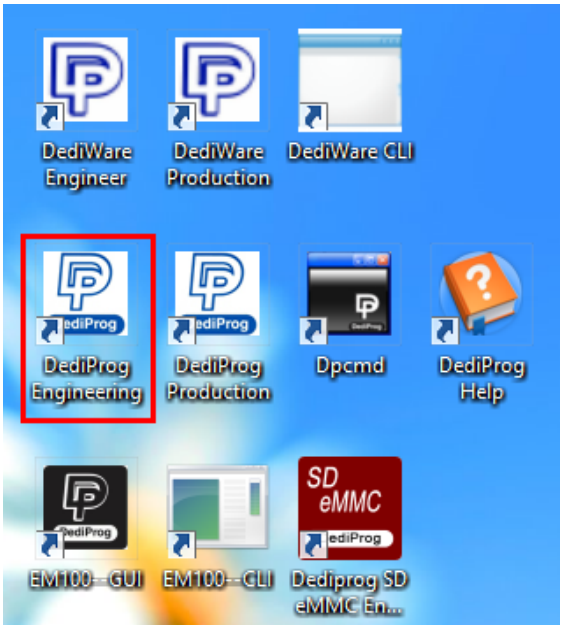
After the software installation, there will be three software icons on your desktop. Icon "DediProg Engineering" is for the engineering GUI, Icon "DediProg Production" is for the production GUI, and Icon "DPCMD" is for the command line interface.

## 3.1 Prepare the Environment

3.1.1 Connect the programmer to the PC through a USB cable.

- For ICP programming, connect the ICP cable to the application (please check the specification in case ISP header pin out are not known).
- For socket and stand alone programming, connect the appropriate socket adaptor to the programmer and insert a serial flash in the socket.

3.1.2 Double click the DediProg software icon on your desktop.



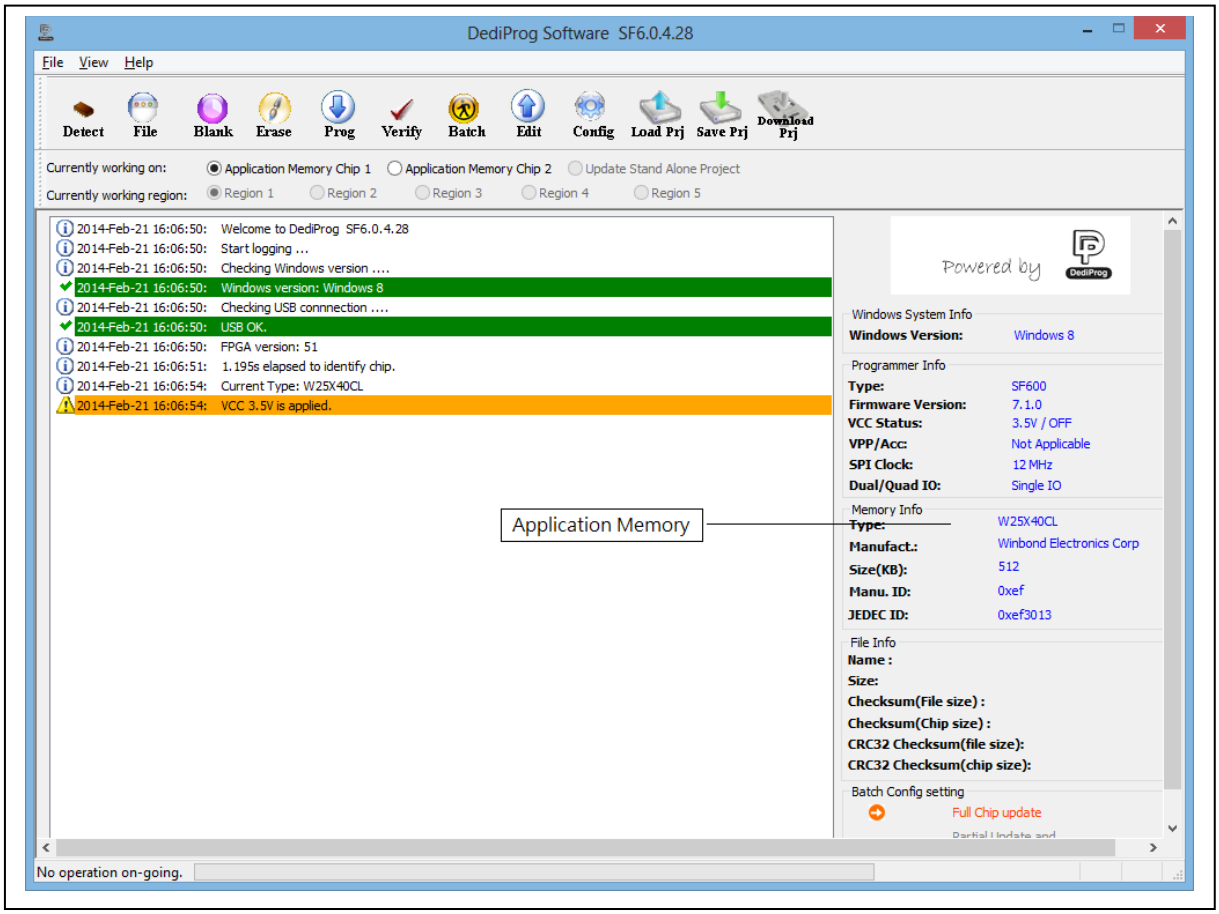
## 3.2 Identify the Target SPI Flash

**SPI Flash Detection**

Double Click the DediProg software icon on your PC desktop. The detected Serial Flash information as well as the programmer information will be displayed on the right side of the window.

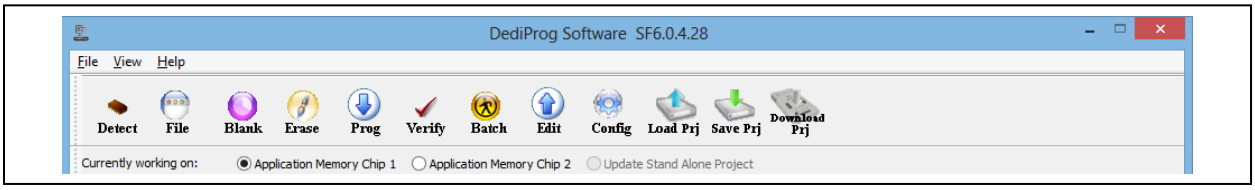
DediProg software will automatically identify the SPI Flash on the application board or socket. User does not need to select SPI Flash' s location.

**Note:** If user wants to work on the second target SPI Flash soldered on the application board, the application board has to be designed with proper schematic and the pin outs have to match with DediProg ISP pin outs.


DediProg Software SF6.0.4.28
File View Help
Detect File Blank Erase Prog Verify Batch Edit Config Load Prj Save Prj Download Prj
Currently working on: Application Memory Chip 1 Application Memory Chip 2 Update Stand Alone Project
Currently working region: Region 1 Region 2 Region 3 Region 4 Region 5
2014-Feb-21 16:06:50: Welcome to DediProg SF6.0.4.28
2014-Feb-21 16:06:50: Start logging ...
2014-Feb-21 16:06:50: Checking Windows version ....
2014-Feb-21 16:06:50: Windows version: Windows 8
2014-Feb-21 16:06:50: Checking USB connnection ....
2014-Feb-21 16:06:50: USB OK.
2014-Feb-21 16:06:50: FPGA version: 51
2014-Feb-21 16:06:51: 1.195s elapsed to identify chip.
2014-Feb-21 16:06:54: Current Type: W25X40CL
2014-Feb-21 16:06:54: VCC 3.5V is applied.
Application Memory
Powered by DediProg
Windows System Info
Windows Version: Windows 8
Programmer Info
Type: SF600
Firmware Version: 7.1.0
VCC Status: 3.5V / OFF
VPP/Acc: Not Applicable
SPI Clock: 12 MHz
Dual/Quad IO: Single IO
Memory Info
Type: W25X40CL
Manufact.: Winbond Electronics Corp
Size(KB): 512
Manu. ID: 0xef
JEDEC ID: 0xef3013
File Info
Name :
Size:
Checksum(File size) :
Checksum(Chip size) :
CRC32 Checksum(file size):
CRC32 Checksum(chip size):
Batch Config setting
Full Chip update
No operation on-going.

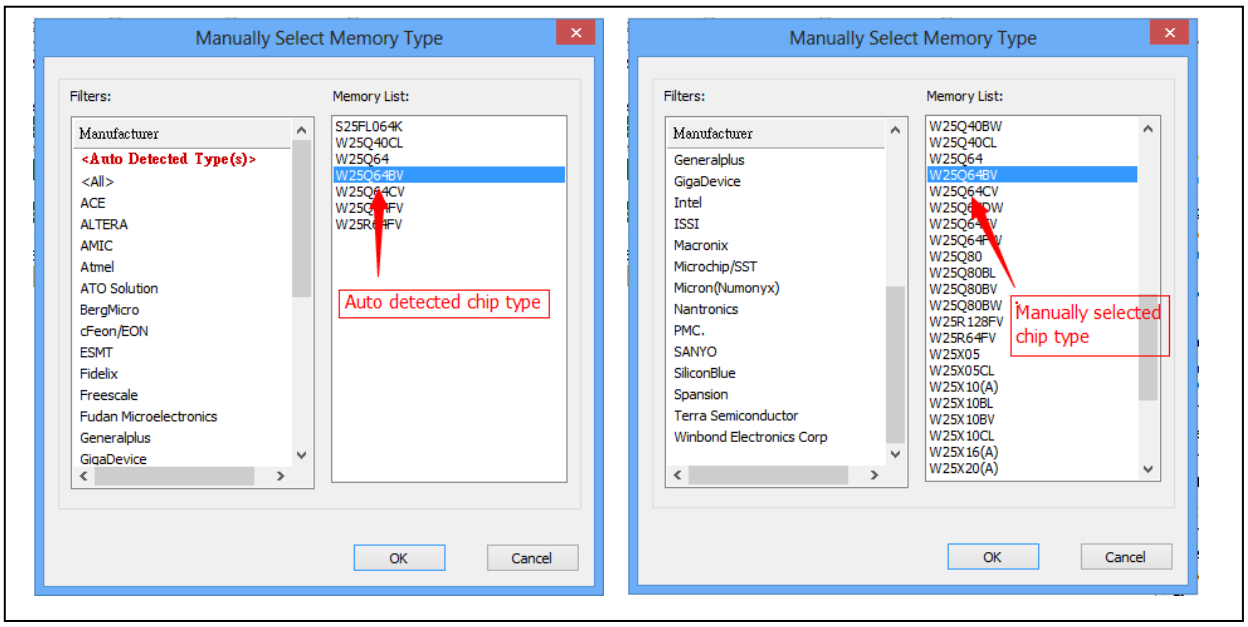
## 3.3 Tool Bar Description

The tool bar provides all SPI Flash operations.



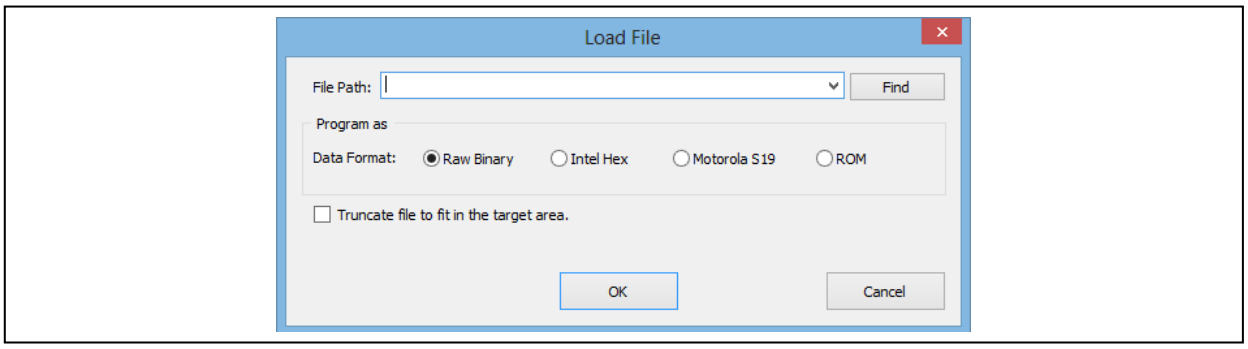

**Detect**

Detect Chip: when a new SPI Flash is placed, user has to click on this button to identify it and perform operations. The auto detected chip types will be displayed on the right side of the screen. In case user would like to manually select a chip type, he/she can move the mouse over the chip manufacturer on the left screen and then click on the chip type on the right screen.




**File**

Select image: load the file you intend to program. The loaded file size cannot be larger the application SPI Flash size.



**Blank**
Blank check: check if the target serial flash is Blank (All Erased)

**Erase**
Erase SPI Flash: Erase the full content in a Serial Flash. After “Erase” the target serial flash shall be blank.

**Prog**
Program: Program the selected image into the Serial Flash

**Verify**
Verify the checksum value of the selected image and the programmed Serial Flash content

**Batch**
Batch operation: The programmer will perform a pre-configured set of operations such as (reload file + erase + program + verify) all together in one click. The configuration can be set by clicking on the “Config” button. The configuration will not be changed until it is re-configured.
Press start button can do batch function when user run the SF software.

**Edit**
When click on Edit, the programmer will by default display the selected file content. User can click on “read” to read and display the chip contents. See “Edit window description” for more details.
**Config**
This allows users to configure advanced settings. See “advanced settings window description” for more details

**Load Prj**
Load the existed project to execute the programming operation.

**Save Prj**
Save all programming settings to a project file for reducing re-setting action.

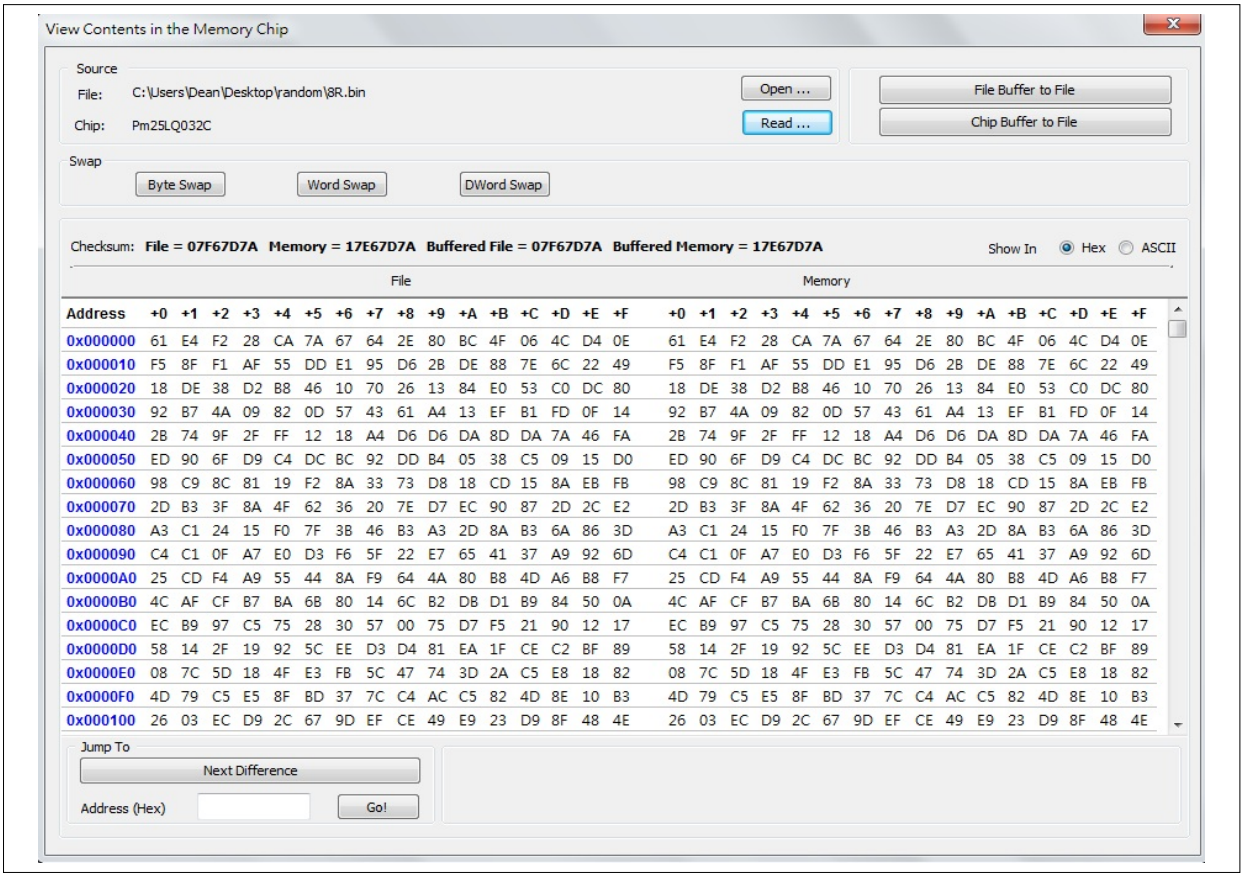
**Download Prj**
SF600*Plus* only, please refer to Chapter 7- **VII. Stand Alone Mode (SF600*Plus* only).**

## 3.4 Edit Window Description

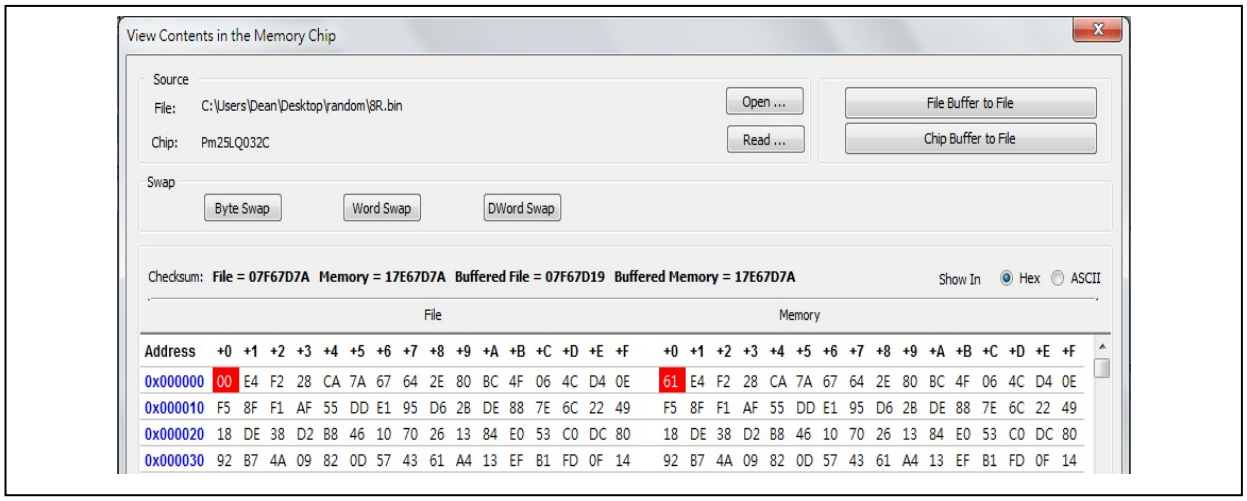
**SPI Flash content display:**

In the edit window, file contents and chip contents can be displayed in the same time so that user can make the comparison. By default the selected file contents are displayed once the user enters into the edit window.

The user can click on “Open” if another file contents are to be shown. The user can click on “Read” in order to read the chip contents are display them on the edit window as well. Checksum of file contents and chip contents are displayed.


View Contents in the Memory Chip
Source
File: C:\Users\Dean\Desktop\random\8R.bin
Chip: Pm25LQ032C
Open ...
Read ...
File Buffer to File
Chip Buffer to File
Swap
Byte Swap
Word Swap
DWord Swap
Checksum: File = 07F67D7A Memory = 17E67D7A Buffered File = 07F67D7A Buffered Memory = 17E67D7A
Show In Hex ASCII
File
Memory
Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
0x000000 61 E4 F2 28 CA 7A 67 64 2E 80 BC 4F 06 4C D4 0E
0x000010 F5 8F F1 AF 55 DD E1 95 D6 2B DE 88 7E 6C 22 49
0x000020 18 DE 38 D2 B8 46 10 70 26 13 84 E0 53 C0 DC 80
0x000030 92 B7 4A 09 82 0D 57 43 61 A4 13 EF B1 FD 0F 14
0x000040 2B 74 9F 2F FF 12 18 A4 D6 D6 DA 8D DA 7A 46 FA
0x000050 ED 90 6F D9 C4 DC BC 92 DD B4 05 38 C5 09 15 D0
0x000060 98 C9 8C 81 19 F2 8A 33 73 D8 18 CD 15 8A EB FB
0x000070 2D B3 3F 8A 4F 62 36 20 7E D7 EC 90 87 2D 2C E2
0x000080 A3 C1 24 15 F0 7F 3B 46 B3 A3 2D 8A B3 6A 86 3D
0x000090 C4 C1 0F A7 E0 D3 F6 5F 22 E7 65 41 37 A9 92 6D
0x0000A0 25 CD F4 A9 55 44 8A F9 64 4A 80 B8 4D A6 B8 F7
0x0000B0 4C AF CF B7 BA 6B 80 14 6C B2 DB D1 B9 84 50 0A
0x0000C0 EC B9 97 C5 75 28 30 57 00 75 D7 F5 21 90 12 17
0x0000D0 58 14 2F 19 92 5C EE D3 D4 81 EA 1F CE C2 BF 89
0x0000E0 08 7C 5D 18 4F E3 FB 5C 47 74 3D 2A C5 E8 18 82
0x0000F0 4D 79 C5 E5 8F BD 37 7C C4 AC C5 82 4D 8E 10 B3
0x000100 26 03 EC D9 2C 67 9D EF CE 49 E9 23 D9 8F 48 4E
Jump To
Next Difference
Address (Hex)
Go!

The difference between file contents and chip contents are highlighted with the “Red Fonts”. User can click on the “next difference” button to search for the next different content between the chip and the file contents.


View Contents in the Memory Chip
Source
File: C:\Users\Dean\Desktop\random\8R.bin
Chip: Pm25LQ032C
Open ...
Read ...
File Buffer to File
Chip Buffer to File
Swap
Byte Swap
Word Swap
DWord Swap
Checksum: File = 07F67D7A Memory = 17E67D7A Buffered File = 07F67D19 Buffered Memory = 17E67D7A
Show In Hex ASCII
File
Memory
Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
0x000000 00 E4 F2 28 CA 7A 67 64 2E 80 BC 4F 06 4C D4 0E 61 E4 F2 28 CA 7A 67 64 2E 80 BC 4F 06 4C D4 0E
0x000010 F5 8F F1 AF 55 DD E1 95 D6 2B DE 88 7E 6C 22 49 F5 8F F1 AF 55 DD E1 95 D6 2B DE 88 7E 6C 22 49
0x000020 18 DE 38 D2 B8 46 10 70 26 13 84 E0 53 C0 DC 80 18 DE 38 D2 B8 46 10 70 26 13 84 E0 53 C0 DC 80
0x000030 92 B7 4A 09 82 0D 57 43 61 A4 13 EF B1 FD 0F 14 92 B7 4A 09 82 0D 57 43 61 A4 13 EF B1 FD 0F 14

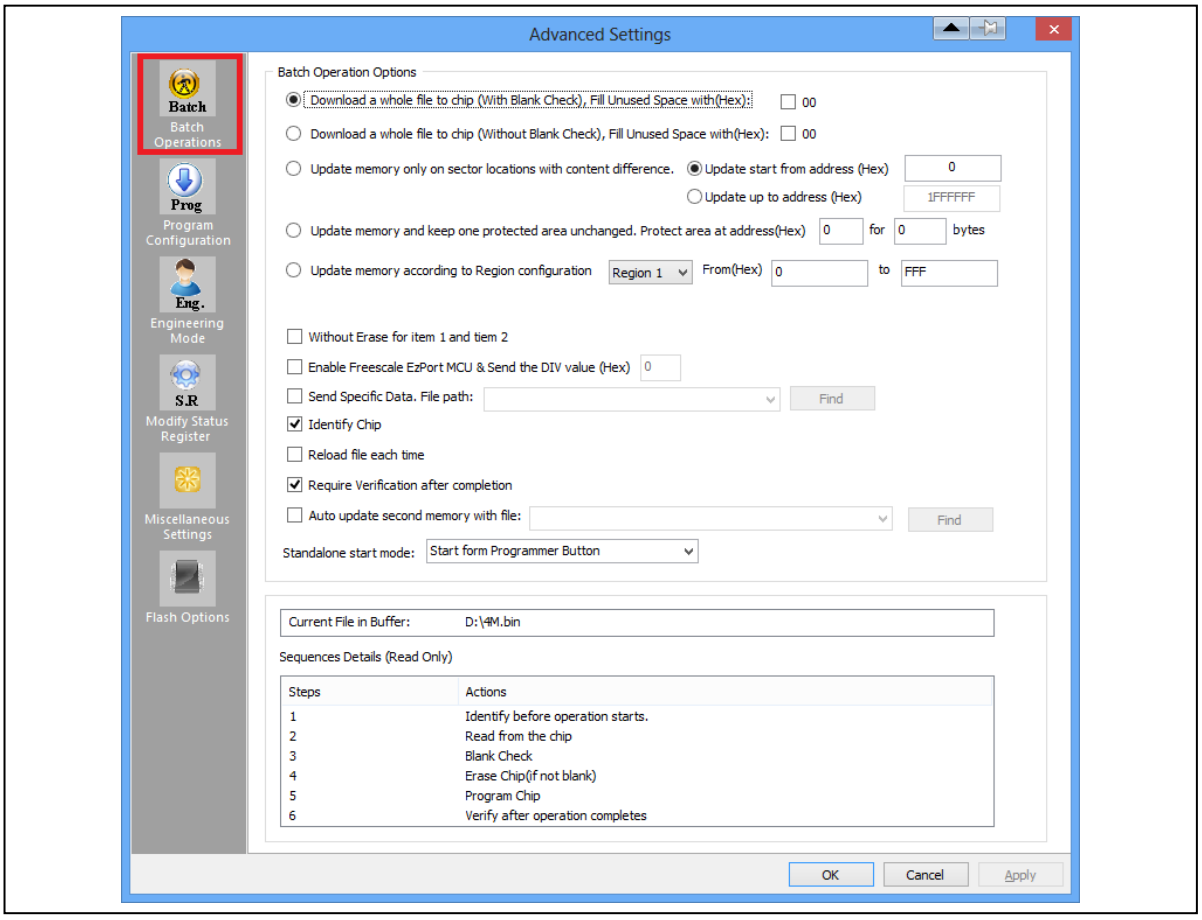
**Chip buffer to file**
This will save the chip contents into a user named binary file.

**File buffer to file**
File buffer can be modified in real time. This button will save the file buffer contents into a user named binary file.

## 3.5 Configuration Window Description

This feature allows users to configure advanced settings

### 3.5.1 Batch Operation Option




A. **Update a Whole file with Blank check**

When user clicks “Batch”, the following operations will be automatically executed:

1) Read the memory content

2) Blank check (check if Chip is erased)

3) Erase the whole memory if not blank

4) Program the whole memory with the file

5) Verify if the memory content is identical with the programmed file.

B. **Update a Whole file without Blank check**

When the user clicks on Batch button, the following operations will be automatically executed:

1) Erase the whole memory

2) Program the whole memory with the file

3) Verify if the memory content is identical with the programmed file.

**C. Update memory only on sector locations with contents difference or Smart update**

User can select the sector locations to have the file programmed.

- **Update start from address (Hex):**

  To program a whole file starting from address 0 of a chip.

- **Update up to address (Hex):**

  To program a whole file, ending at the last address of a chip. The default ending address will automatically calculated by the software according to memory size.

When the user clicks on Batch button, the following operations will be automatically executed:

1) Read the memory content

2) Compare the memory content from the given address with the file at the 64KB sector base

3) Erase only the 64KB sectors with some differences

4) Program only the erased sectors with the file data of the corresponding address

5) Verify the data on the updated 64KB sectors

**Smart Update can be used in the following cases:**

- A small file can be programmed or updated at a given address without any change on the rest of the memory (local update).
- A file with only minor change compare to the memory content can be quickly updated. The sectors without difference are kept unchanged.

**Remark:**
The file data which are identical with the target memory but with an address shift (after compilation) will be interpreted as different and will not benefit of the Smart update advantages.

**D. Update memory and keep one protected area unchanged**

When the user clicks on Batch button, the following operations will be automatically executed:

1) Read the memory content from the given address for the given length

2) Insert the read memory contents into the file buffer

3) Erase the whole chip

4) Program the whole chip with the updated file in step 2

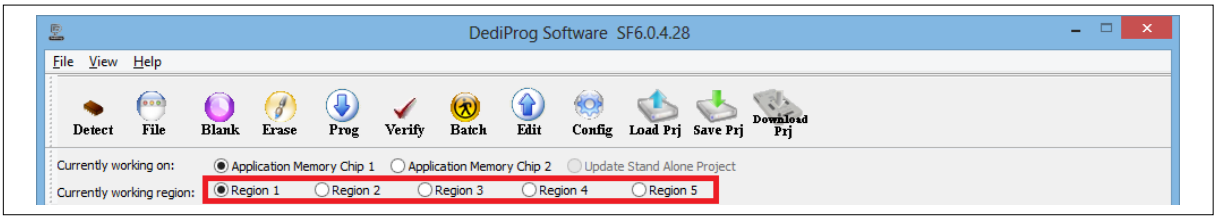
5) Verify the programmed data

**E. Update memory according to Region configuration**

Sometimes user only wants to update some part of the data in SPI Flash. User can use this function to update the data in the assigned region. This function saves time when debugging.

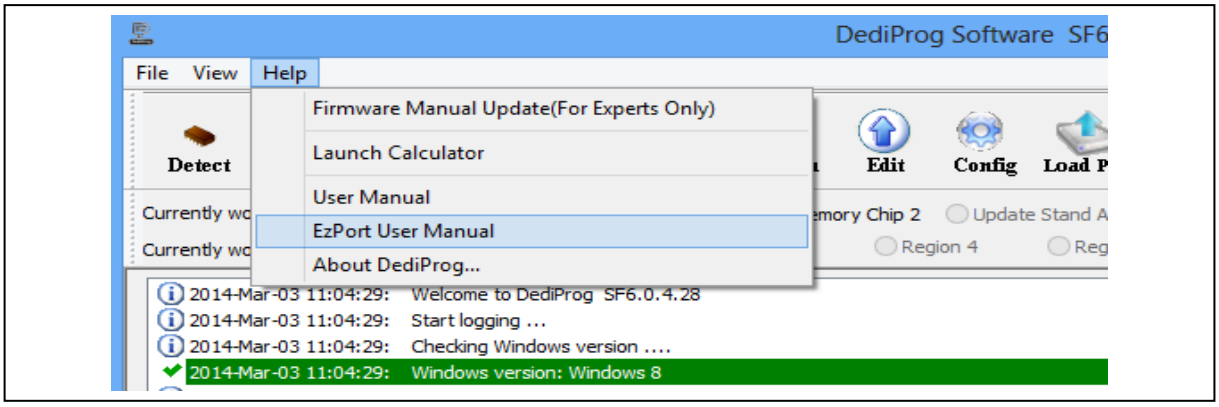
1) Assign the Region and set start & end address of the Region.

2) Select working region


DediProg Software SF6.0.4.28
File View Help
Detect File Blank Erase Prog Verify Batch Edit Config Load Prj Save Prj Download Prj
Currently working on: Application Memory Chip 1 Application Memory Chip 2 Update Stand Alone Project
Currently working region: Region 1 Region 2 Region 3 Region 4 Region 5

**F. Enable Freescale EzPort MCU & Send the DIV value (Hex)**

If the box is checked, the programmer will automatically enable EzPort. Details please see the « Help →EzPort User Manual»




**G. Send Specific Data**

If the box is checked, the software will load and send the engineering SPI sequence defined and saved in the “Engineering Mode” Configuration window. This option allows user to create his/her own SPI instruction.

**H. Identify Chip**

If the box is checked, the software will identify before operation starts.

**I. Reload file each time**

If the box is checked, the software will load the same file from the source destination each time before the batch operations (refresh). This option is helpful when another software update the file in parallel (like compiler).

**J. Require Verification after completion**

If this box is checked, the software will verify the contents between the source file and the programmed Serial Flash contents after the batch operations.

## Methods Comparison:

### Case 1:

64Mb Serial flash update with 64Mb file totally different. Memory has been previously programmed and need to be totally erased.

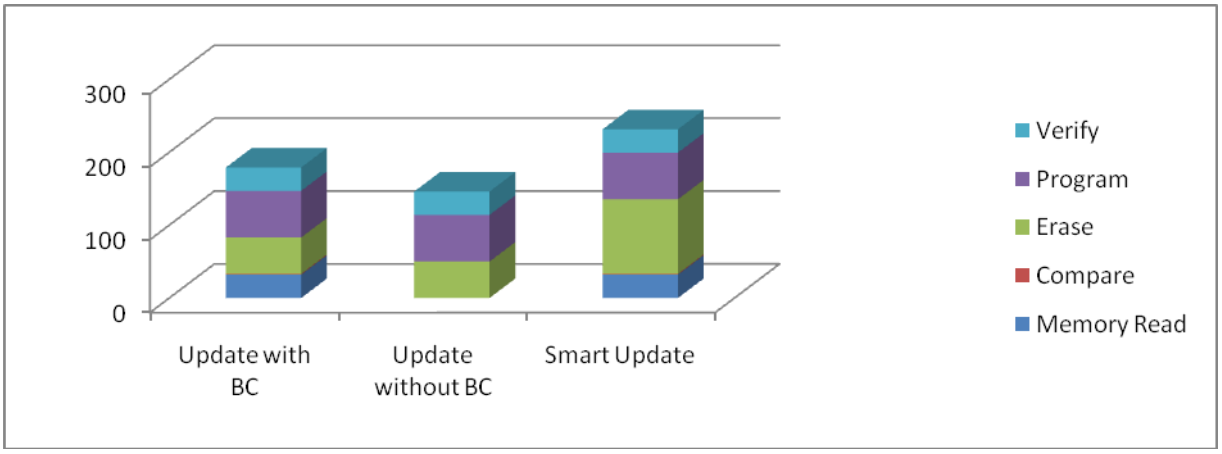
Chip Erase: 50sec

Sector Erase (64KB): 0.8sec

Chip: 128 sectors

| | Update with BC | Update without BC | Smart Update |
|---|---|---|---|
| **Memory Read** | **32 sec** | **none** | **32 sec** |
| **Compare** | **1 sec** | **none** | **1 sec** |
| **Erase** | **50 sec** | **50 sec** | **100 sec** |
| **Program** | **64 sec** | **64 sec** | **64 sec** |
| **Verify** | **32 sec** | **32 sec** | **32 sec** |
| **TOTAL** | **179 seconds** | **146 seconds** | **229 seconds** |

### Comparison Chart




### Conclusion:

If the memory needs to be completely Erased for a file update, the “Update without Blank Check” is the optimum choice.

### Time Saving: 20%

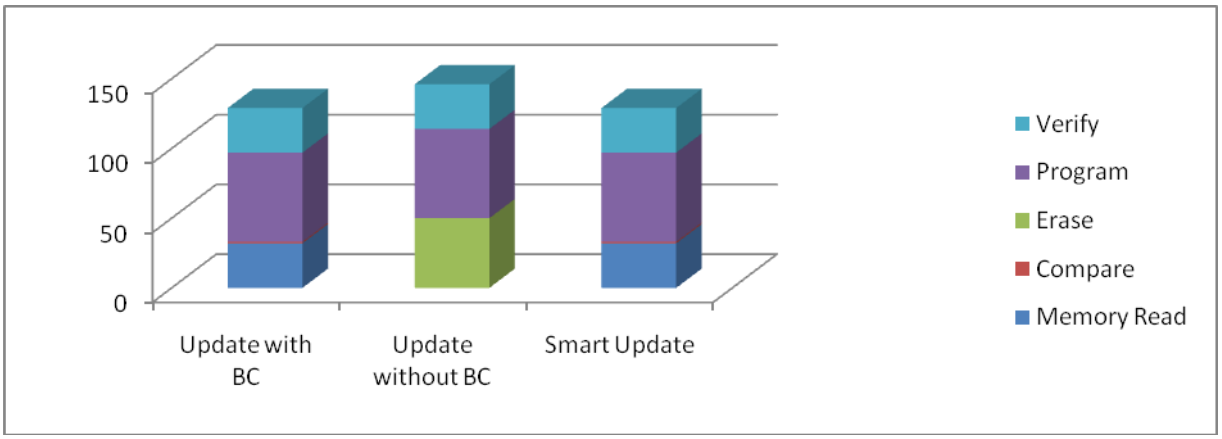
## Case 2:

64Mb Serial flash programming with a 64Mb file.

Memory has never been programmed (from supplier).

| | Update with BC | Update without BC | Smart Update |
|---|---|---|---|
| **Memory Read** | **32 sec** | **none** | **32 sec** |
| **Compare** | **1 sec** | **none** | **1 sec** |
| **Erase** | **0 sec** | **50 sec** | **0 sec** |
| **Program** | **64 sec** | **64 sec** | **64 sec** |
| **Verify** | **32 sec** | **32 sec** | **32 sec** |
| **TOTAL** | **129 seconds** | **146 seconds** | **129 seconds** |

## Comparison Chart




## Conclusion:

If the memory is blank (from supplier), the “Update with Blank Check” or “Smart update” is the optimum choice.

## Time Saving: 12%

## Case 3:

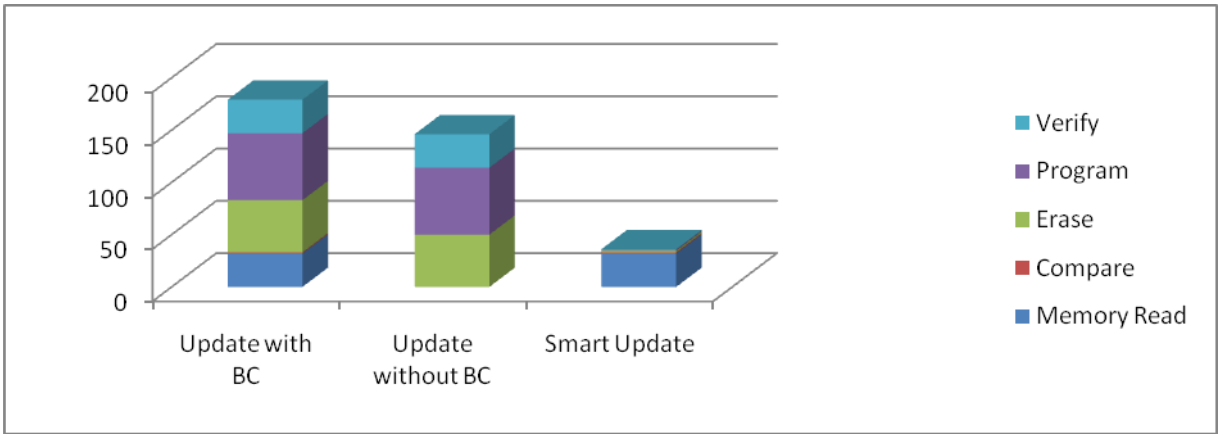
64Mb Serial flash update with a 64Mb file with only data differences on 2 sectors or a small file of 2 sectors size only at a specified address.
Sector Erase: 0.8 sec

Sector Programming: 0.5 sec

Sector Verify: 0.25 sec

| | Update with BC | Update without BC | Smart Update |
|---|---|---|---|
| **Memory Read** | **32 sec** | **none** | **32 sec** |
| **Compare** | **1 sec** | **none** | **1 sec** |
| **Erase** | **50 sec** | **50 sec** | **1.6 sec** |
| **Program** | **64 sec** | **64 sec** | **1 sec** |
| **Verify** | **32 sec** | **32 sec** | **0.5 sec** |
| **TOTAL** | **179 seconds** | **146 seconds** | **36 seconds** |

## Comparison Chart

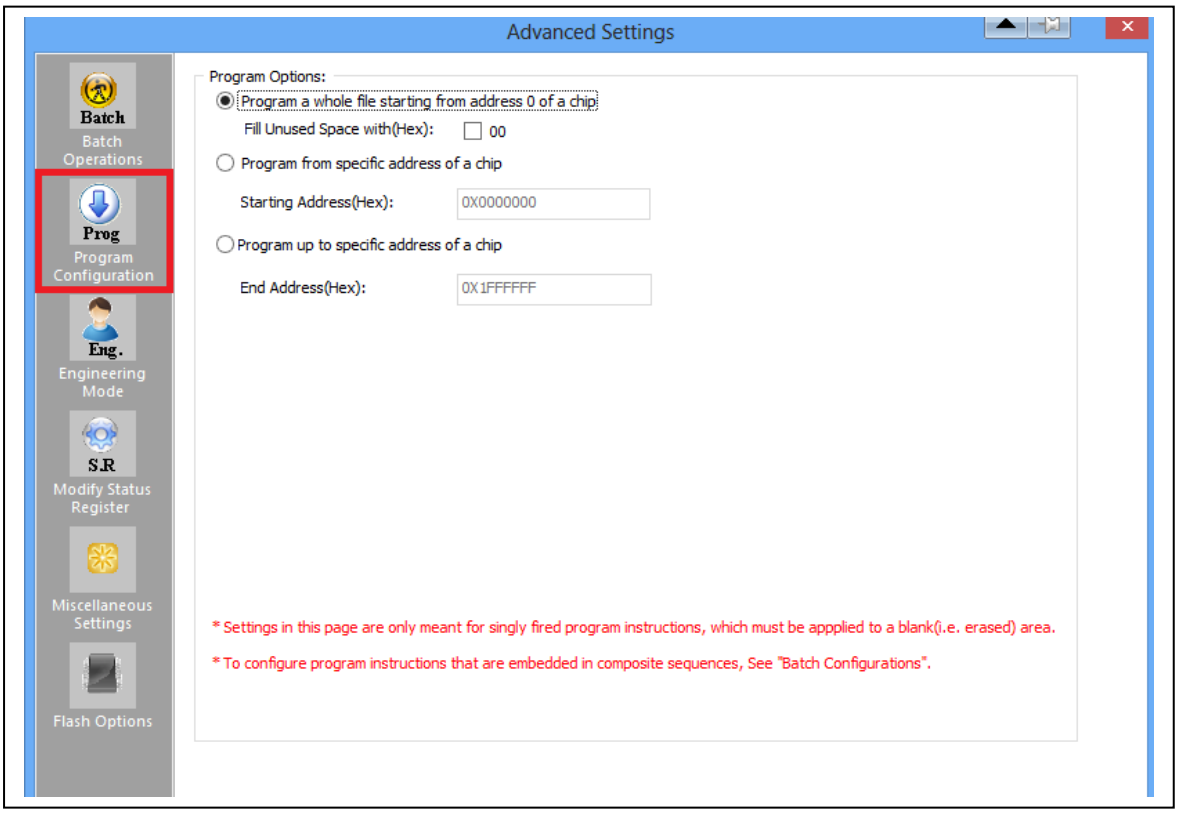



## Conclusion:

If the difference between the memory content and file are small or if the file to be programmed is small, the “Smart update” is the optimum choice.

## Time Saving: 75%

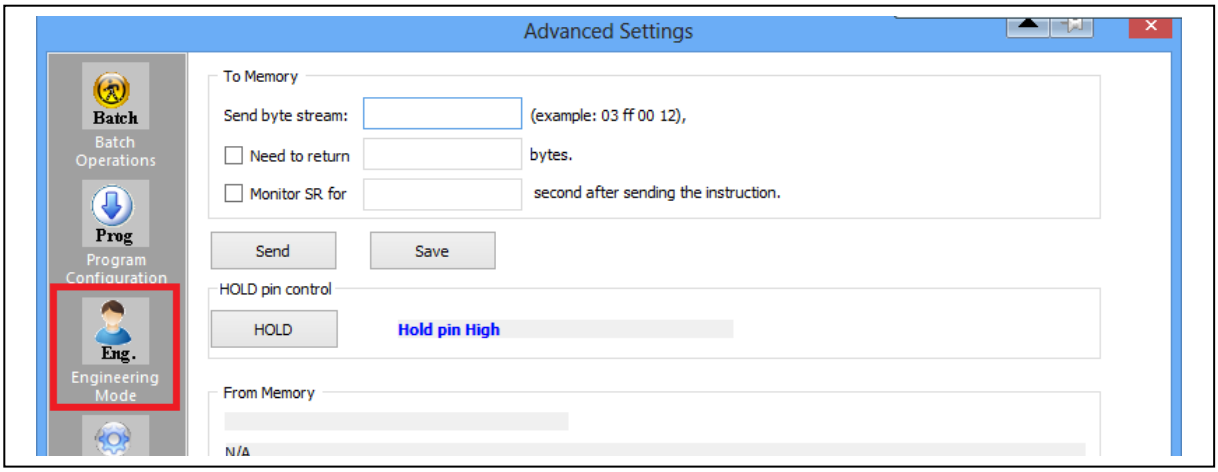
### 3.5.2 Program Configurations


Advanced Settings
Batch
Batch Operations
Prog
Program Configuration
Eng.
Engineering Mode
S.R
Modify Status Register
Miscellaneous Settings
Flash Options
Program Options:
Program a whole file starting from address 0 of a chip
Fill Unused Space with(Hex):
00
Program from specific address of a chip
Starting Address(Hex):
0X0000000
Program up to specific address of a chip
End Address(Hex):
0X1FFFFFF
* Settings in this page are only meant for singly fired program instructions, which must be appplied to a blank(i.e. erased) area.
* To configure program instructions that are embedded in composite sequences, See "Batch Configurations".

A. Program a whole file starting from address 0 of a chip
B. Program from specific address of a chip: To program a whole file starting from address 0 of a chip.
C. Program up to specific address of a chip: To program a whole file, ending at the last address of a chip. The default ending address will automatically calculated by the software according to memory size.

If the file is smaller than the target Serial Flash, user can define how to fill the rest of the SPI Flash. By default FFh or 00h if selected in the interface.

### 3.5.3 Engineering Mode



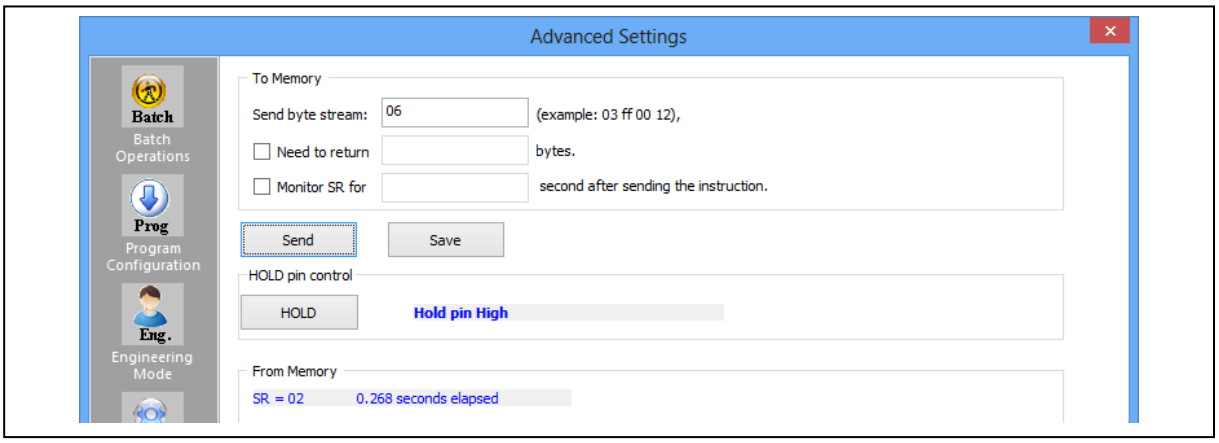

This function allows users to define their own SPI command and send it directly to the target SPI flash. This option is powerful as all the not standard SPI commands can be generated even if not supported by our programmer.

Users can define the data bytes to be sent from the programmer to the SPI Flash and the number of bytes to be returned. Users can also define if the status register WIP bit has to be polled to check if the SPI Flash is busy or ready.

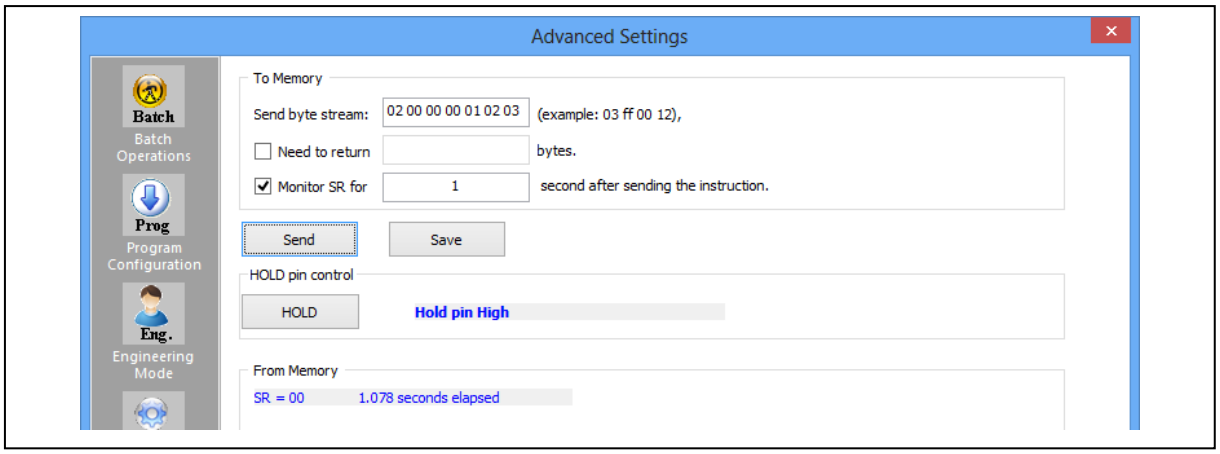
Users can save the stream data for future use by click on the “Save” button.

**For example:**
User wants to write “01 02 03” data bytes at the address “00 00 00” and verify.

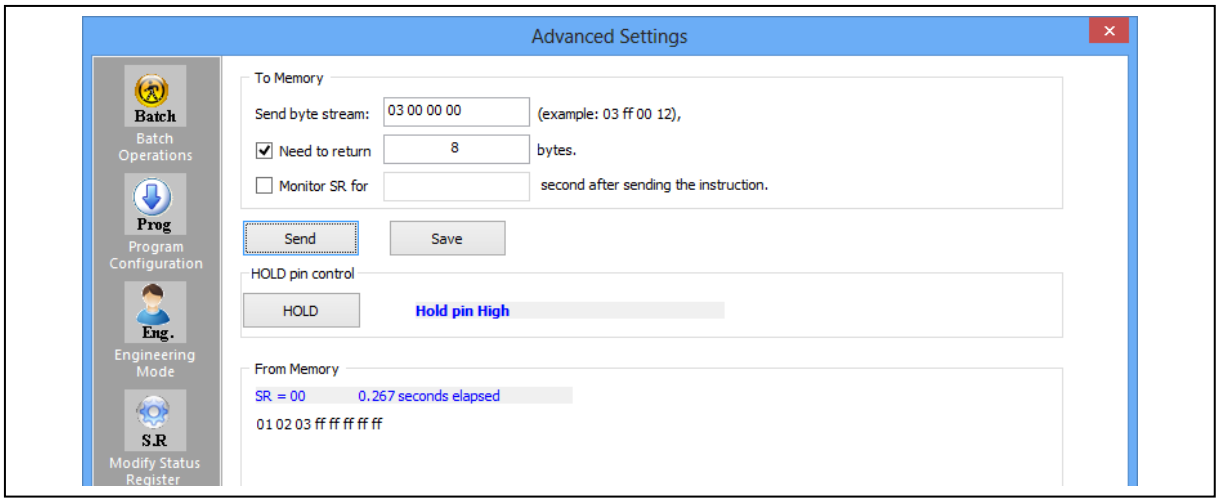
**First:** programmer needs to set the WEL bit by sending the WREN (06h) command to the SPI Flash as described below:



**Second:** programmer needs to send the programming instruction “02h” followed by the address “00 00 00” and the data “01 02 03” and monitor the Status register WIP bit as described below:



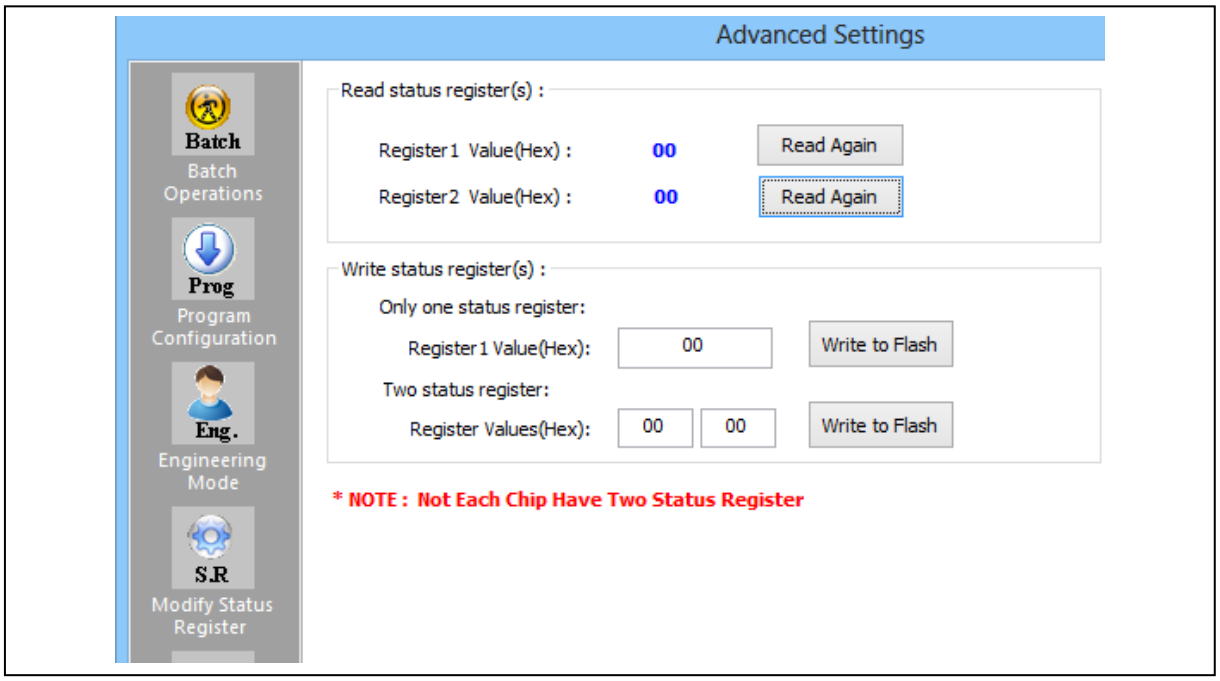

**Third:** The programmer need to verify the SPI Flash content by sending the Read instruction “03h” and the address “00 00 00” then read the return bytes from the SPI Flash (we read 8 bytes in the following example):




The return bytes from the SPI Flash are displayed in the “from SPI Flash” window: “01 02 03 FF FF FF FF FF”.

The engineering mode can be used to send any instruction to the SPI Flash.

### 3.5.4 Modify Status Register




This function allows users to modify or read the status register(s) value of the target serial flash.

Please note each chip has their own command to write status registers.

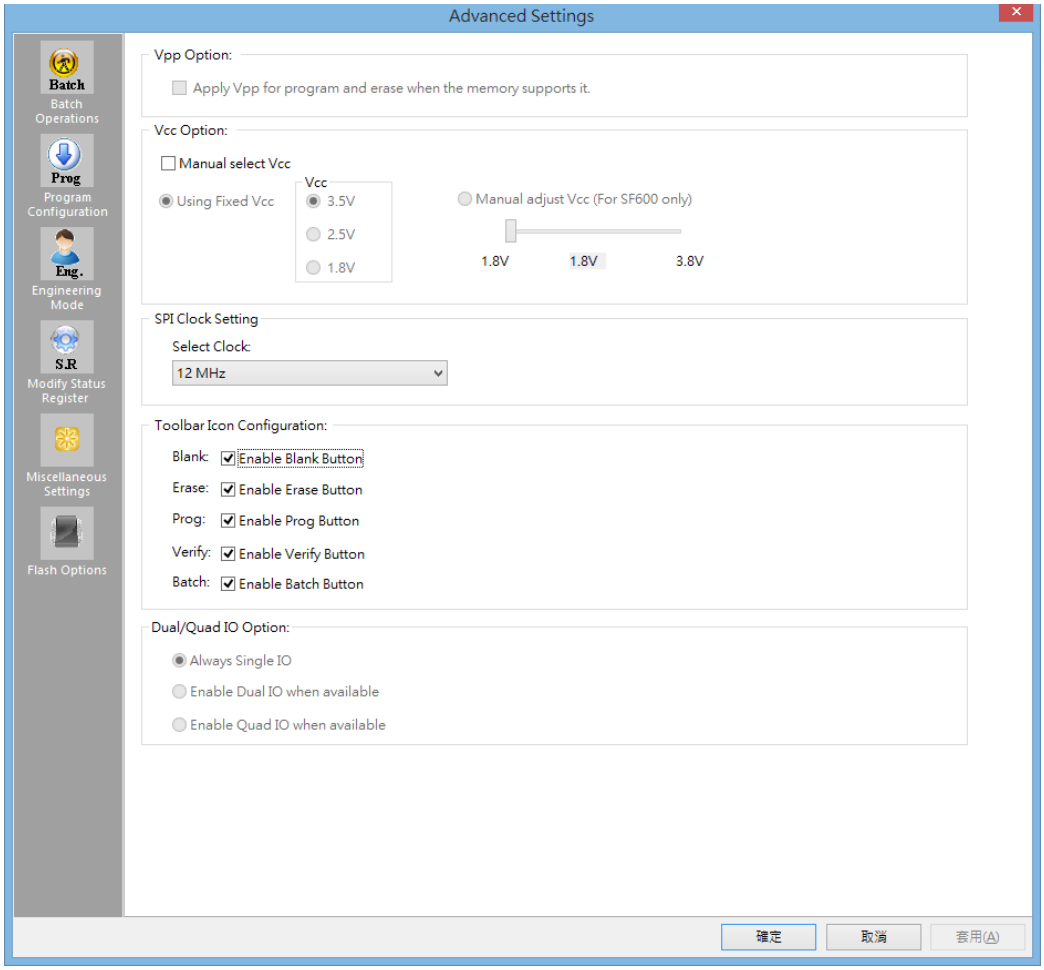
For chip only has one status register:

- For write: “06h” to set the Write Enable and “01h” and user data” to write the status register
- For Read: “05h” to read the status register

For chip has two status registers:

- Please refer to the device specification for parameter setting.

## 3.5.5 Miscellaneous Settings

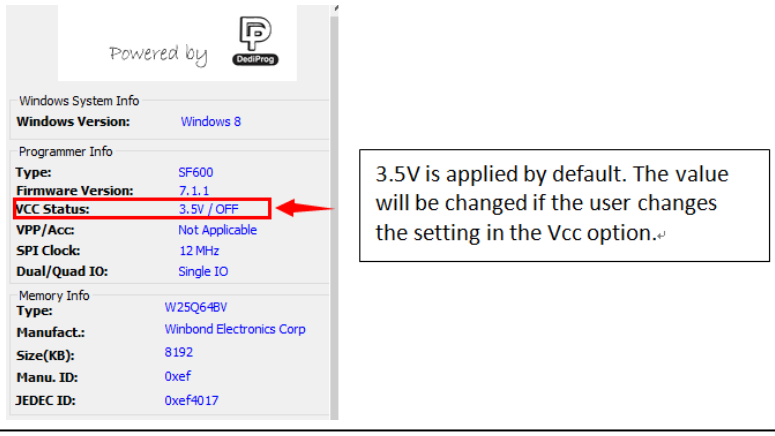



### A. Vpp Option

This setting allows user to enable the Vpp option so the High voltage is applied on the SPI Flash Wp pin to reduce the programming and erasing time.
This option can only be enabled on Serial Flash supporting the Vpp feature.

### B. Vcc Option

SF series programmers support 3.5V, 2.5V, and 1.8V Vcc. Default of 3.5V Vcc is applied after the software installation. User is able to change the Vcc configuration here and the Vcc setting will be changed and saved until next modification.



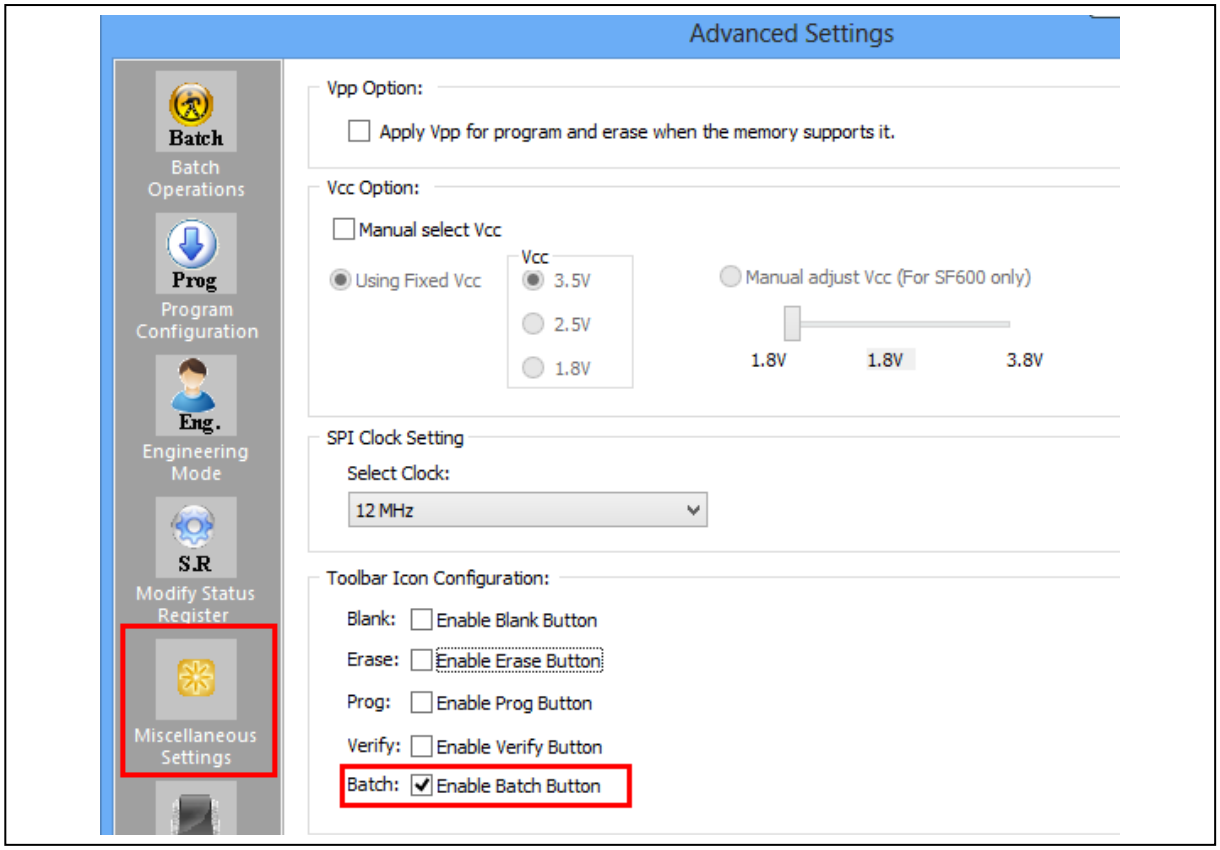
### C. SPI Clock Setting

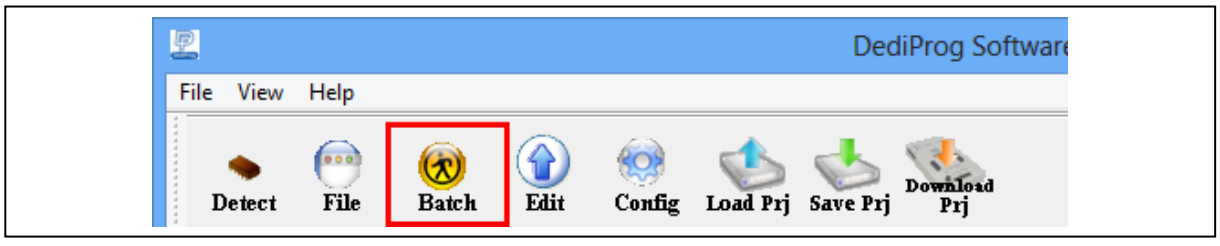
The SPI clock frequency can be adjusted by user to fit the application requirements or SPI Flash performance. Notice that the SPI Flash frequency is defined in the supplier specification for a maximum capacitance usually of 30pf or 15pF max. The application is therefore designed to not exceed this maximum capacitance. In circuit programming does not fulfill anymore this original design as additional capacitance will be added according to the cable length and programmer. Therefore, user cannot expect to program the on board SPI flash according to the maximum frequency of the datasheet as the SPI flash will not be able to drive such capacitance at such high frequency.

In order to comply with the different capacitance and SPI flash driving capability, DediProg provides frequency adjustment of the programmer. Frequency needs to be reduced if the data timings do not comply with the specification.

### D. Tool Bar ICON Configuration

Users can hide some tool bar icons if they unselect the icon items in the “tool bar icon configuration setting”. For example, if the engineer only wants the operators to use batch icon, he/she can leaves only batch icon selected and save the setting. The operators will only see the batch icon on the tool bar.

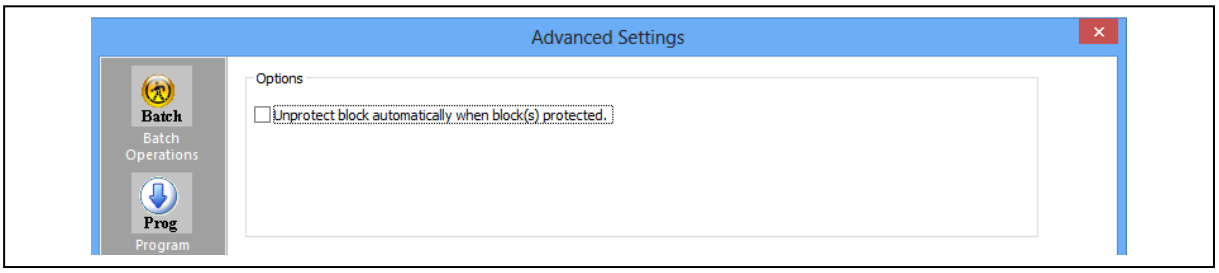


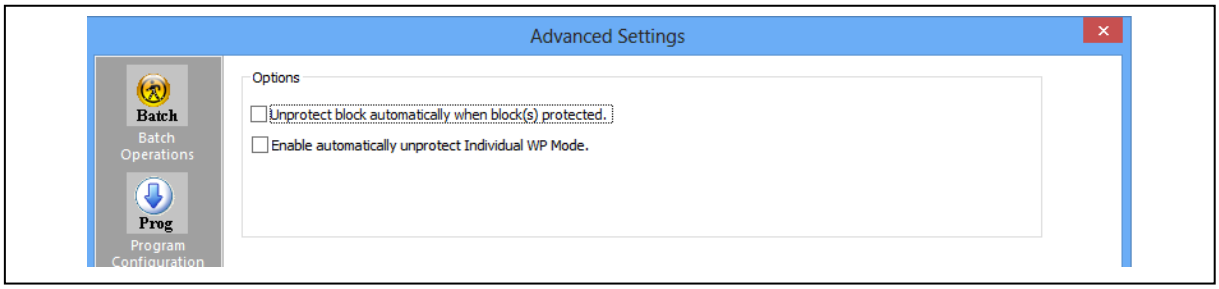
### 3.5.6 Flash Option

There're three kinds of options.

A. Unprotect block automatically when block(s) protected.

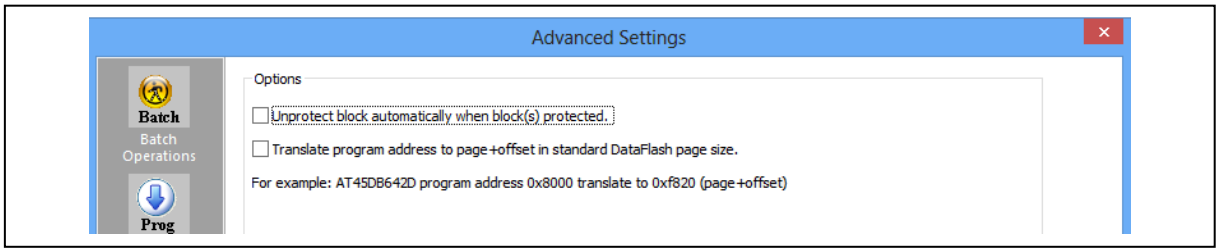



B. Enable automatically unprotect Individual WP mode



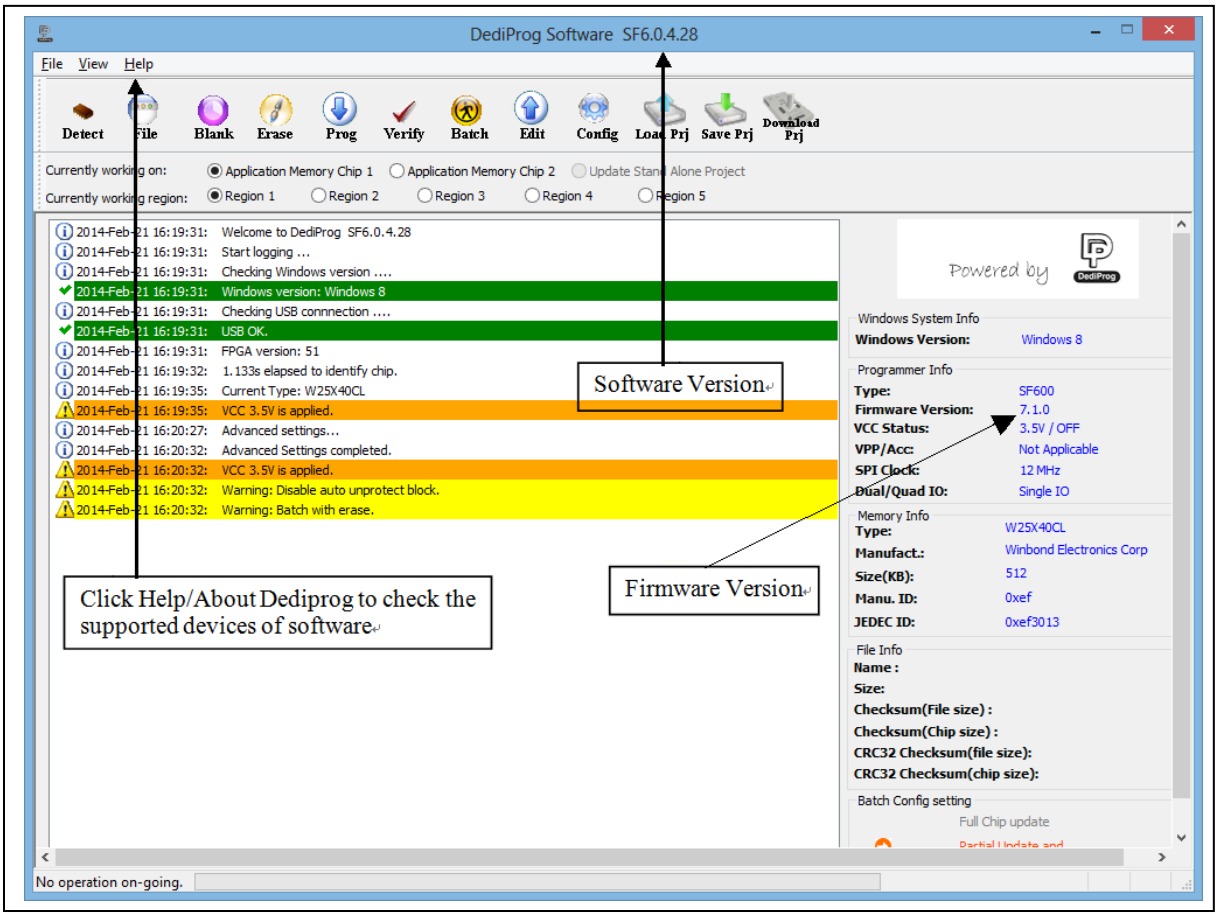

C. Translate program address to page+offset in standard DataFlash page size.

For example: AT45DB642D program address 0x8000 translate to 0xF820 (page+offset)


Advanced Settings
Batch
Batch Operations
Prog
Options
Unprotect block automatically when block(s) protected.
Translate program address to page+offset in standard DataFlash page size.
For example: AT45DB642D program address 0x8000 translate to 0xf820 (page+offset)

## 3.6 Supported Devices, Software Version, Firmware Version

User can check the Serial flash support list in our web site. This support list is valid for the latest software and firmware so user will have to check the current software and firmware version he is using and update it if necessary.


DediProg Software SF6.0.4.28
File View Help
Detect File Blank Erase Prog Verify Batch Edit Config Load Prj Save Prj Download Prj
Currently working on: Application Memory Chip 1 Application Memory Chip 2 Update Stand Alone Project
Currently working region: Region 1 Region 2 Region 3 Region 4 Region 5
2014-Feb-21 16:19:31: Welcome to DediProg SF6.0.4.28
2014-Feb-21 16:19:31: Start logging ...
2014-Feb-21 16:19:31: Checking Windows version ....
2014-Feb-21 16:19:31: Windows version: Windows 8
2014-Feb-21 16:19:31: Checking USB connnection ....
2014-Feb-21 16:19:31: USB OK.
2014-Feb-21 16:19:31: FPGA version: 51
2014-Feb-21 16:19:32: 1.133s elapsed to identify chip.
2014-Feb-21 16:19:35: Current Type: W25X40CL
2014-Feb-21 16:19:35: VCC 3.5V is applied.
2014-Feb-21 16:20:27: Advanced settings...
2014-Feb-21 16:20:32: Advanced Settings completed.
2014-Feb-21 16:20:32: VCC 3.5V is applied.
2014-Feb-21 16:20:32: Warning: Disable auto unprotect block.
2014-Feb-21 16:20:32: Warning: Batch with erase.
Software Version
Firmware Version
Click Help/About Dediprog to check the supported devices of software
Powered by DediProg
Windows System Info
Windows Version: Windows 8
Programmer Info
Type: SF600
Firmware Version: 7.1.0
VCC Status: 3.5V / OFF
VPP/Acc: Not Applicable
SPI Clock: 12 MHz
Dual/Quad IO: Single IO
Memory Info
Type: W25X40CL
Manufact.: Winbond Electronics Corp
Size(KB): 512
Manu. ID: 0xef
JEDEC ID: 0xef3013
File Info
Name :
Size:
Checksum(File size) :
Checksum(Chip size) :
CRC32 Checksum(file size):
CRC32 Checksum(chip size):
Batch Config setting
Full Chip update
No operation on-going.

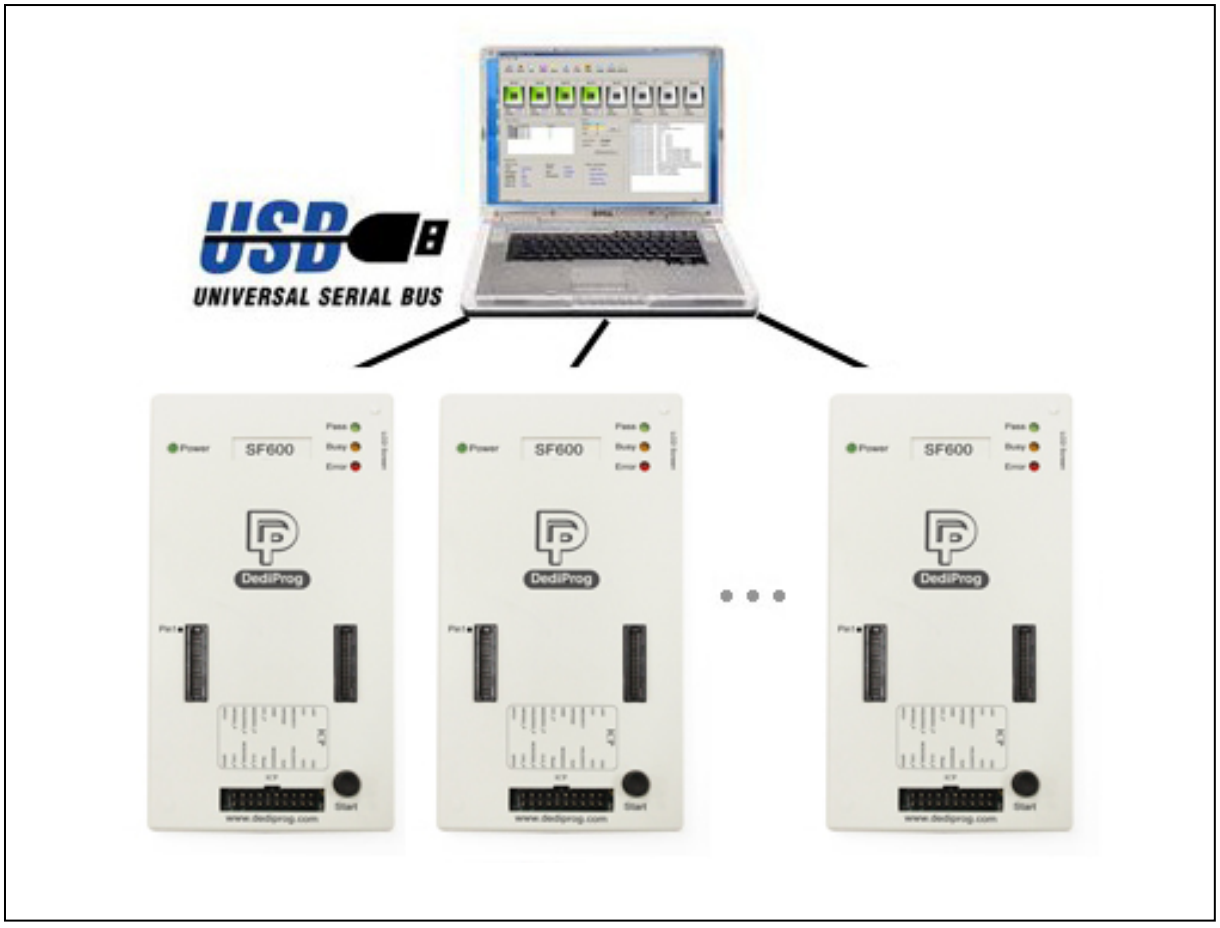
# IV. DediProg SF Software Production GUI

DediProg SF software production GUI is only available after the software version 5.x.x. The production GUI allows users to plug in and operate multiple SF100/SF600/SF600*Plus* in the same time.

The new software will remove the old USB driver when it detects such driver during the installation. New USB driver is required in order to run the software and the driver will come together with the software CD ROM or it can be downloaded from DediProg website. **www.dediprog.com/download**

In order to run more than one SF programmer in the same time reliably, USB hub with individual power supply is highly recommended.

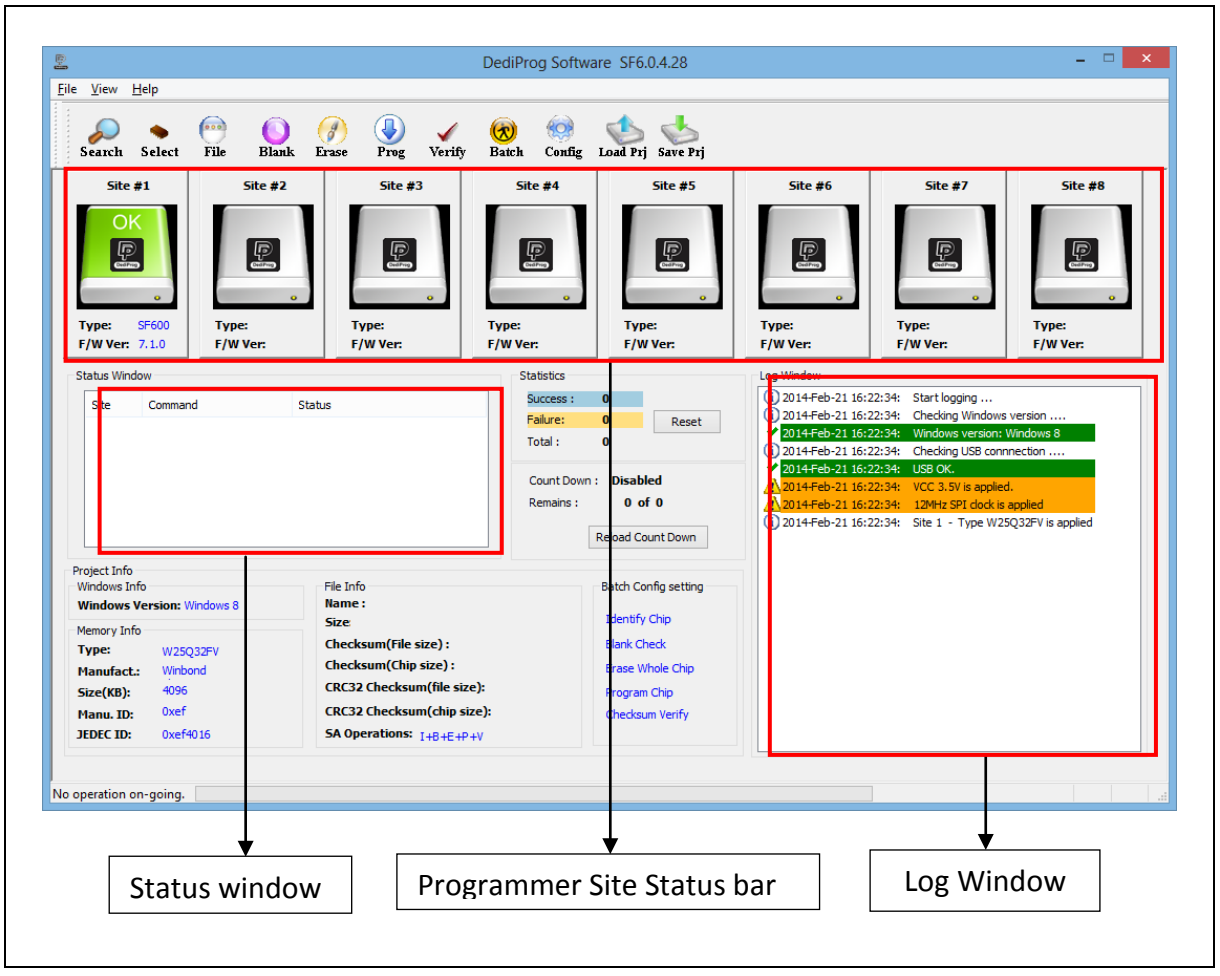
Multi-Programmers Capability for SF series programmers



In order to run production GUI, USB plug in of all the intended programmers is required prior to opening the software. It is not recommended to add (plug in) or reduce (unplug) any number of programmers when the software is already opened.

The production software does not provide auto chip detect feature and therefore “programmer search” and “chip select” are required prior to any other operations.

The production GUI manual will only illustrate the items not covered by the engineering GUI. Therefore function explanations such as Program, Erase, Blank check, etc will not be repeated here.

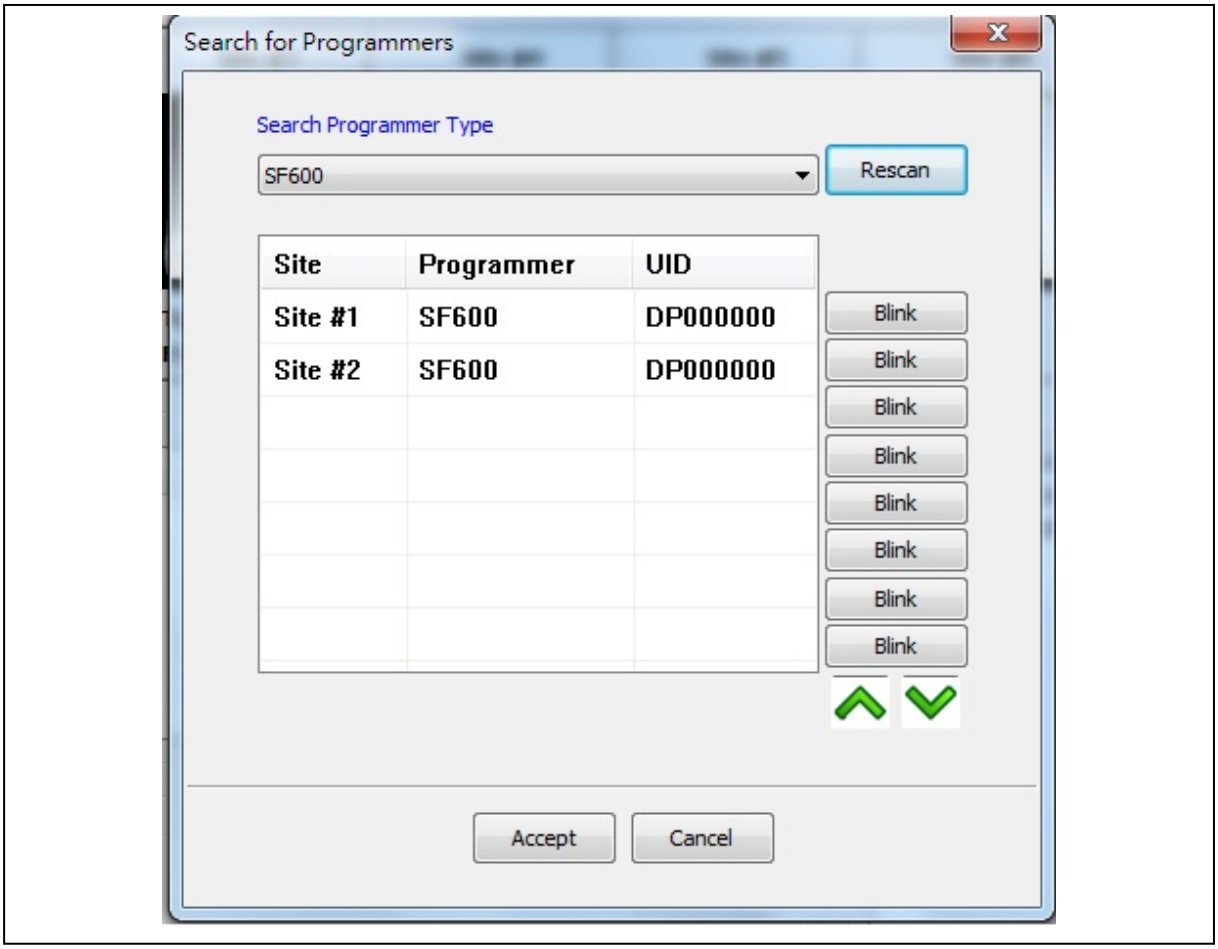


## 4.1 Search and Select

When click on “search”, the software will show programmer type. The default of programmer type is SF100. Please select the programmer you are using and click Rescan.

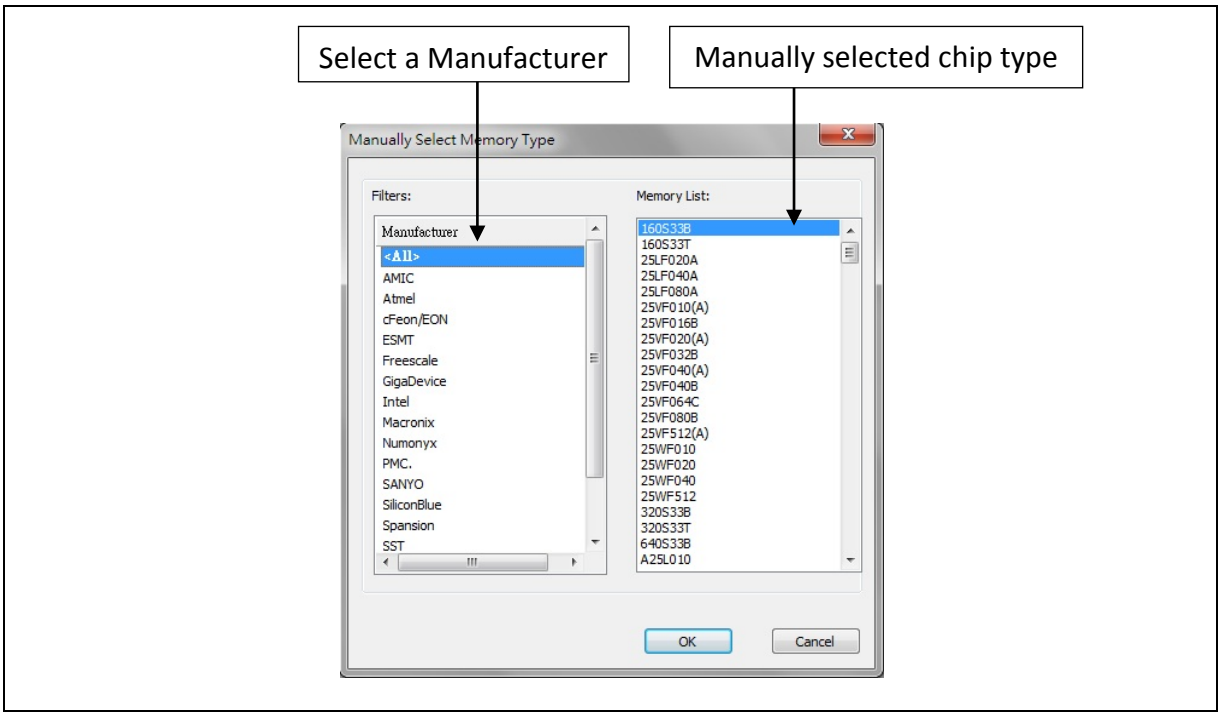
**Search Programmer:**

The found programmers will be listed along with site number. The site number is given by the Window OS randomly and therefore users can use the “blink”, “up” and “down” button to adjust the real sequence of the connected programmer. When click on “blink”, the connected programmer will blink on its green LED once. Users can use this feature to locate the programmer associated with its site number. For programmers with firmware version after 5.x.x, DediProg will write a serial number in the hardware before shipping out and the serial number will be displayed in the following screen snapshot.

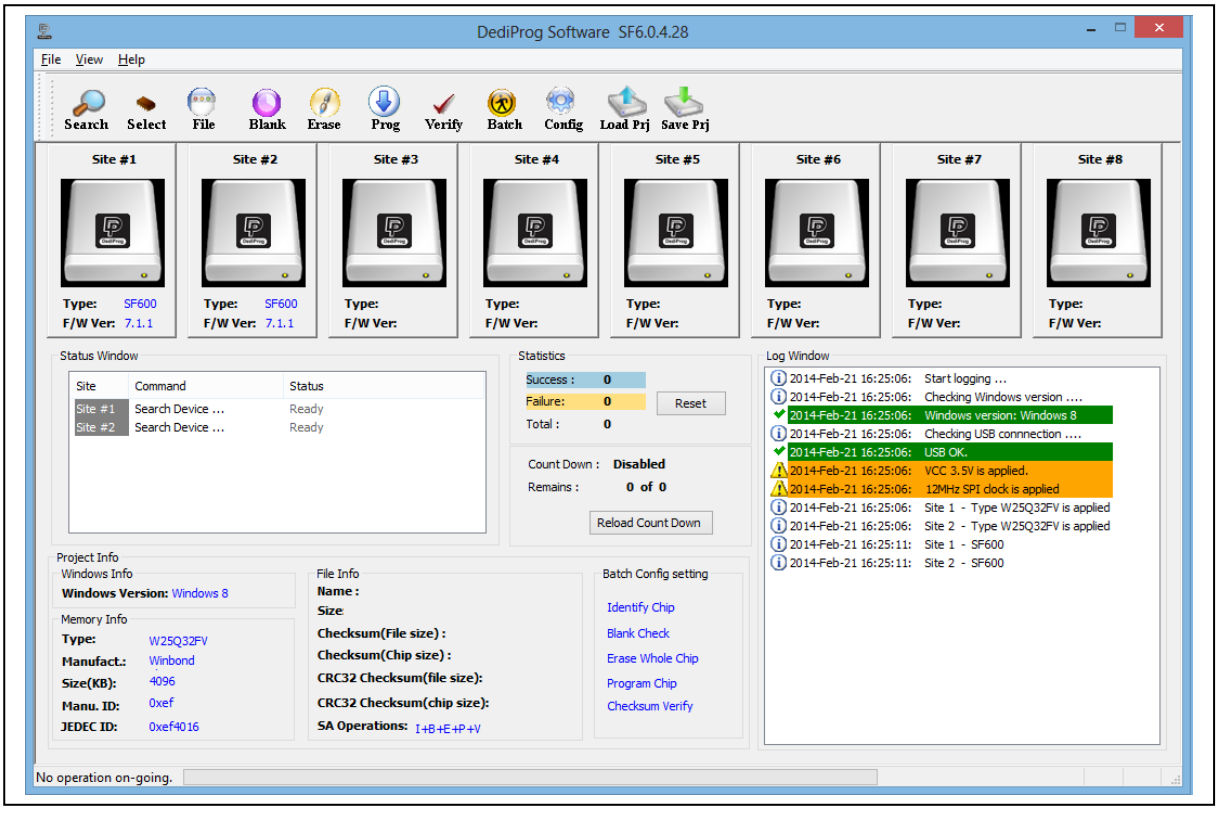


**Select Memory Type:**

The production software does not provide auto chip detect feature users will need to select the target memory manually.

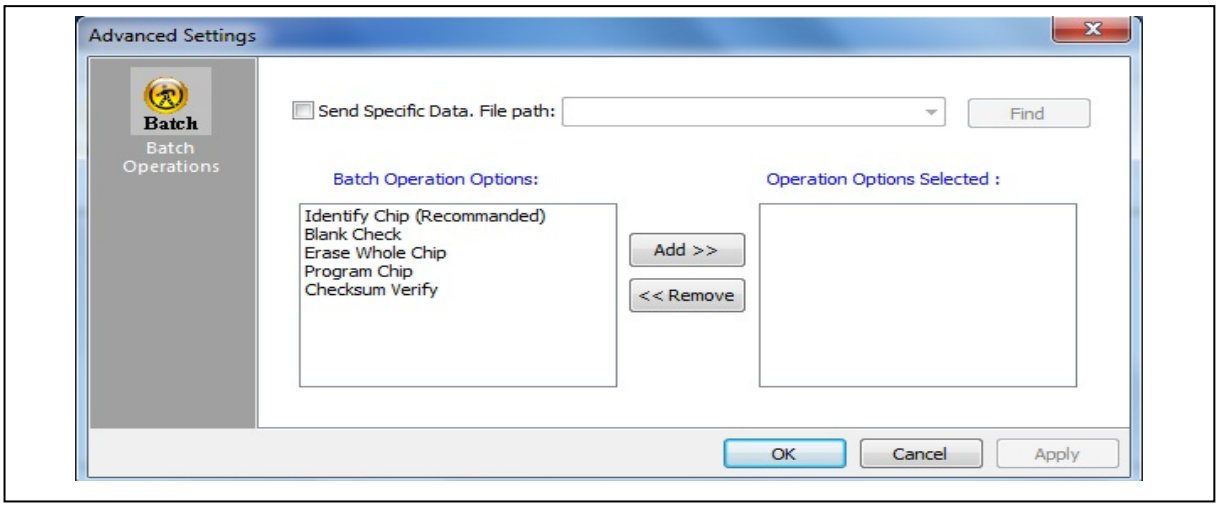



After the search step and the to-be-programmed chip is selected, the main GUI will have updated information on the Programmer SITE Status bar, the status window and the log window.



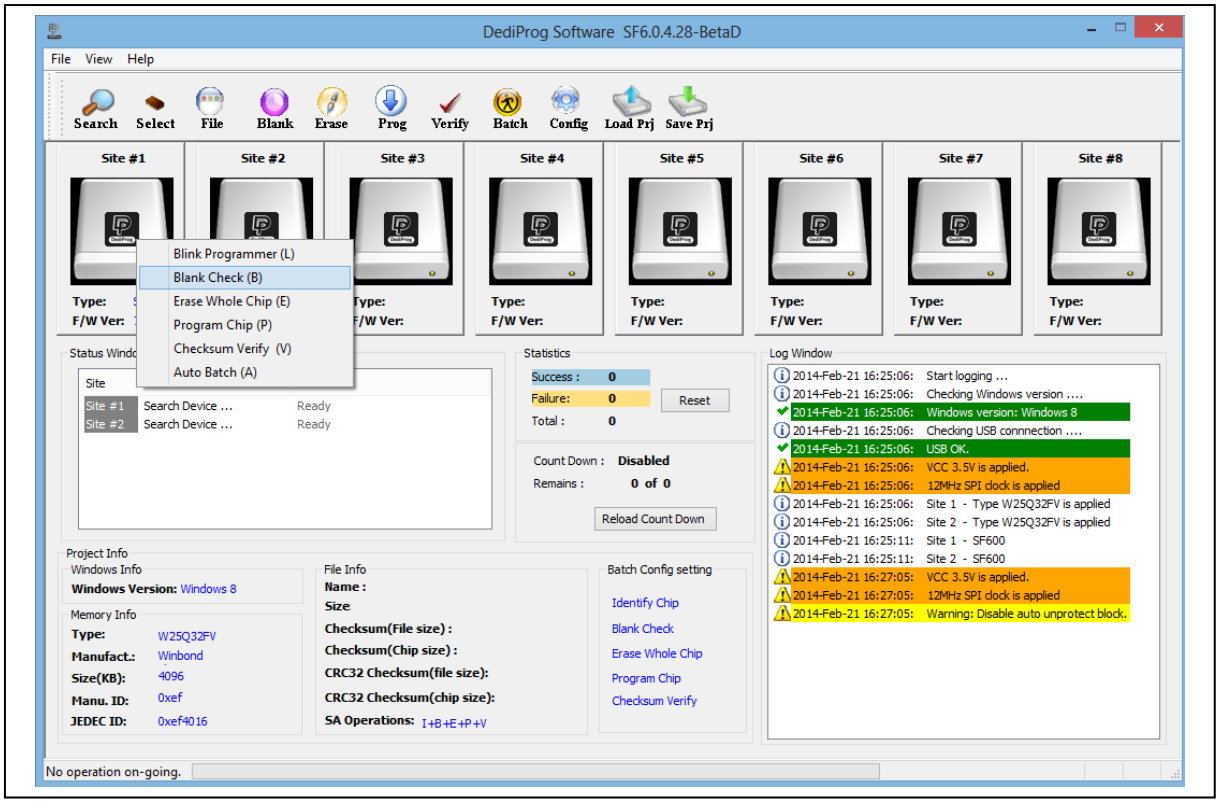
## 4.2 Batch Config

By clicking on the “Config” icon, users can access to configure the batch setting.
Users may click on the option “Send Specific Data” for sending the stream data before reading/writing the device. This customized SPI sequence can be created in the “engineering interface”.
Users may click on the “Add” or “Remove” for Batch Operation Options directly.




## 4.3 Single Site programming

By click on the right mouse button after pointing to a specific Programmer Site number, users will have the access of programming options to the pointed programmer site.



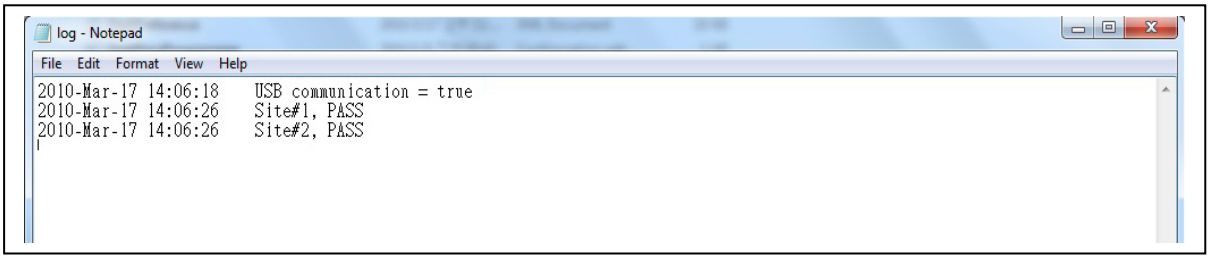
# V.DediProg Windows Command Line

## 5.1 Introduction

The window command line has been designed to control our programmer from another software. This feature will be convenient to synchronize the two software in development (For example: program the memory automatically after the code has been compiled) or in production (for example: Program automatically the Serial Flash via the ICT tester after the hardware has been checked).

Command result “log.txt” file will be automatically saved under following folders:
Windows XP:
C:\Documents and Settings\User\Application Data\DediProg\SF100

Windows Vista, Windows 7, Windows 8 and Windows 8.1:
C:\Users\user\AppData\Roaming\DediProg\SF100


log - Notepad
File Edit Format View Help
2010-Mar-17 14:06:18 USB communication = true
2010-Mar-17 14:06:26 Site#1, PASS
2010-Mar-17 14:06:26 Site#2, PASS

This .txt file has to be checked to make sure that the operation has been successful. Time stamp can also be checked to be sure that the result has been updated with a new value.

Below are the error messages in the log.txt file.
FAIL Identify Fail
FAIL Blank Fail
FAIL Erase Fail
FAIL Program Fail
FAIL Read Fail
FAIL Send Specific data Fail
FAIL Verify Fail
FAIL Unknow

To get more information about these methods please contact DediProg.

## Window DOS command

```
Basic Usages:
Dpcmd -uxxx
Dpcmd /uxxx
Dpcmd --auto=xxx
(space is not needed between the switches and parameters. E.g. dpcmd -ubio.bin)

Basic Switches(switches in this group are mutual exclusive):
  -? [ --help ]                 show this help message
  --list                        print supported chip list
  -d [ --detect ]               detect chip
  -b [ --blank ]                blank check
  -e [ --erase ]                erase entire chip
  -r [ --read ] arg             read chip contents and save to a bin/hex/s19
                                file
                                - use STDOUT for the console.
  -p [ --prog ] arg             program chip without erase
  -u [ --auto ] arg             automatically run the following sequence:
                                - Read the memory content
                                - Compare the memory content
                                - Erase only the sectors with some differences
                                - Program only the erased sectors with the file
                                data from address 0
  -z [ --batch ] arg            automatically run the following sequence:
                                - check if the chip is blank or not;
                                - erase the entire chip(if not blank);
                                - program a whole file starting from address 0
  -s [ --sum ]                  display chip content checksum
  -f [ --fsum ] arg             display the file checksum
                                - needs to work with a file
  --raw-instruction arg         issue raw serial flash instructions.
                                - use spaces(" ") to delimit bytes.
                                - instructions must be enclosed in double
                                quotation marks("")
                                Example:
                                dpcmd --raw-instruction "03 FF 00 12"
  --raw-require-return arg (=0) decimal bytes of result to return in decimal
                                after issuing raw instructions.
                                - used along with --raw-instruction only.
                                Example:
                                dpcmd --raw-instruction "03 FF 00 12" --raw-req
                                uire-return 1

Optional Switches that add fine-tune ability to Basic Switches:
  -a [ --addr ] arg       hexadecimal starting address hexadecimal(e.g.
                          0x1000),
                          - works with --prog/read/sum/auto only
                          - defaults to 0, if omitted.
  -l [ --length ] arg     hexadecimal length to read/program in bytes,
                          - works with --prog/read/sum/auto only
                          - defaults to whole file if omitted
  -v [ --verify ]         verify checksum file and chip
                          - works with --prog/auto only
  -x [ --fill ] arg (=FF) fill spare space with an hex value(e.g.FF),
                          - works with --prog, --auto only
  --type arg              Specify a type to override auto detection
                          - use --list arguement to look up supported type.
```

```
  --lock-length arg       hexadecimal length of area that will be kept
                          unchanged while updating
                          - used along with --auto only.
  --blink arg
                          - 0 : Blink green LED 3 times from USB1 to USBn
                          (Default)
                             note: the sequence is assigned by OS during USB
                          plug-in
                          - 1: Blink the programmer connected to USB1 3 times.
                          - n: Blink the programmer connected to USBn 3 times.
  --device arg            (work with all Basic Switches)
                          - 1: activate only the programmer connected to USB1
                          - n: activate only the programmer connected to USBn
                             note: if "--device" is not used, the command will
                          be executed with the same chip type and file on all
                          connected programmer.
  --fix-device arg        Fix programmer serial number with programmer
                          sequence.
                          - instructions must be enclosed in double quotation
                          marks("")
                          Example:
                          dpcmd --fix-device "1 DP000001"
  --list-device-id arg
                          - 0 : List all ID of programmers from USB1 to USBn
                          (Default)
                             note: the sequence is assigned by OS during USB
                          plug-in
                          - 1: Prompt the device ID of programmer connected to
                          USB1.
                          - n: Prompt the device ID of programmer connected to
                          USBn.

Miscellaneous options:
  -t [ --timeout ] arg (=300) Timeout value in seconds
  -g [ --target ] arg (=1)    Target Options
                              Available values:
                                  1, Chip 1(Default)
                                  2, Chip 2
                                  3, Socket
                                  0, reference card
  --vcc arg (=4)              specify vcc
                                  0, 3.5V(Default)
                                  1, 2.5V
                                  2, 1.8V
                                  1800 ~ 3800, 1.8 ~ 3.8V (minimum step 100mV)
                              (For SF600 only)
                              - work with --prog and --erase.
  --vpp                       apply vpp when the memory chip supports it
  --log                       write operation result into file "%appdata%\dedip
                              rog\SF100\log.txt"
  -i [ --silent ]             suppress the display of real-time timer counting
                              - used when integrating with 3rd-party tools(e.g.
                              IDE)
  --spi-clk arg (=2)          specify SPI clock:
                                  2, 12 MHz(Default)
                                  0, 24 MHz
                                  1, 8 MHz
                                  3, 3 MHz
                                  4, 2.18 MHz
                                  5, 1.5 MHz
                                  6, 750 KHz
                                  7, 375 KHz
```

## 5.2 How to Start

DediProg window dos command line software is executed by the file “dpcmd.exe.” There are three different ways to run the dos command line.

1. Double click on the “dpcmd” icon on your desktop and type in dpcmd and enter.
2. Change your dos directory to the same location where “dpcmd.exe” is located. C:\program files\dediprog\SF100
3. Type in the following command to auto directs the dpcmd command to the “dpcmd.exe” location.

**Set path=%path%;”c:\program files\dediprog\SF100**

## 5.3 Basic Usages

1. dpcmd –r "f:\file.bin",
   reads the chip and save it into a file "file.bin" in Partition f
2. dpcmd –r STDOUT –a 0x100 -l 0x23,
   reads 0x23 bytes starting from 0x100 and display it on the screen
3. dpcmd –u f:\file.bin,
   erases and then program file.bin in Partition f into the serial flash
4. dpcmd –p f:\file.bin –a 0x100,
   writes file.bin in Partition f into the serial flash starting from address 0x100
5. dpcmd –p f:\file.bin –x 0xaa,
   programs file.bin in Partition f into the serial flash and fill the rest area with 0xaa

**Remarks:** -a, -l only works with -p, -r, -s
**Remarks:** -x only works with -p
**Remarks:** space is not needed between the switches parameters. E.g. dpcmd –uf:\file.bin

## 5.4 Basic Switches

| | |
|---|---|
| -? [ --help ]<br>--list<br>-d [ --detect ]<br>-b [ --blank ]<br>-e [ --erase ]<br>-r [ --read ] arg<br><br>-p [ --prog ] arg<br>-u [ --auto ] arg<br>- Erase only the sectors | -z [ --batch ] arg<br>Show this help message<br>Print supproted chip list<br>detect chip<br>blank check<br>erase entire chip<br>read chip contents and save to a bin/hex/s19 file<br>use STDOUT for the console.<br>program chip without erase<br>utomatically run the following sequence:<br>- Read the memory c<br>- Compare the memory content references<br>- Program only the erased sectors with the file data from address 0<br>automatically run the following sequence: |

| | |
|---|---|
| -s [ --sum ]<br>-f [ --fsum ] arg<br><br>--raw-instruction arg<br><br><br><br><br>--raw-require-return arg (=0) | - check if the chip is blank or not;<br>- erase the entire chip(if not blank);<br>- program a whole file starting from address 0<br>display chip content checksum<br>display the file checksum<br>- needs to work with a file<br>issue raw serial flash instructions.<br>- use spaces(" ") to delimit bytes.<br>- instructions must be enclosed in double quotation marks("")<br>Example:<br>dpcmd --raw-instruction "03 FF 00 12"<br>decimal bytes of result to return in decimal after issuing raw instructions.<br>- used along with --raw-instruction only.<br>Example:<br>dpcmd --raw-instruction "03 FF 00 12"<br>raw-require-return 1 |

## 5.5 Optional Switches

**(Specify the following switches to change default values):**

| | |
|---|---|
| -a [ --addr ] arg<br><br><br>-l [ --length ] arg<br><br>-v [ --verify ]<br><br>-x [ --fill ] arg (=FF)<br><br>--type arg<br><br>hexadecimal starting address hexadecimal (e.g. 0x1000),<br>- works with --prog/read/sum/auto only<br>- defaults to 0, if omitted.<br>hexadecimal length to read/program in bytes,<br>- works with -- | - works with --prog, --auto only<br>Specify a type to override auto detection<br>- use --list arguement to look up supported type.<br><br>--lock-start arg<br><br>--lock-length arg<br><br>--blink arg<br><br>--device arg |

| | |
|---|---|
| prog/read/sum/auto only<br>- defaults to whole file if omitted<br>verify checksum file and chip<br>- works with --prog/auto only<br>fill spare space with an hex value(e.g. FF),<br><br>hexadecimal starting address(e.g. 0x1000),<br>- works with --prog/read/sum/auto only<br>- defaults to 0, if omitted.<br>hexadecimal length of area that will be kept unchanged while updating<br>- used along with --auto only.<br>- 0 : Blink green LED 3 times from USB1 to USBn (Default)<br>note: the sequence is assigned by OS during USB plug-in<br>- 1: Blink the programmer connected to USB1 3 times. | --fix-device arg<br><br><br><br><br><br>--list-device-id arg<br><br><br><br>- n: Blink the programmer connected to USBn 3 times.<br>(work with all Basic Switches)<br>- 1 : activate only the programmer connected to USB1<br>- n : activate only the programmer connected to USBn<br>note: if "--device" is not used, the command will be executed with the same chip type and file on all connected programmer.<br>Fix programmer serial number with programmer sequence.<br>- instructions must be enclosed in double quotation marks("")<br>Example:<br>dpcmd --fix-device "1 DP000001"<br>- 0 : List all ID of programmers from USB1 to USBn (Default)<br>note: the sequence is assigned by OS during USB plug-in<br>- 1 : Prompt the device ID of programmer connected to USB1.<br>- n : Prompt the device ID of programmer connected to USBn. |

**Miscellaneous options:**

Note: The programming operation always uses the default value for command. If users want to use other setting, must add the wanted option to every command.

| | |
|---|---|
| -t [ --timeout ] arg (=300)<br>-g [ --target ] arg (=1) | Timeout value in seconds<br>Target Options<br>Available values:<br>1, Chip 1(Default)<br>2, Chip 2<br>3, Socket<br>0, reference card |
| --vcc arg (=0) | specify vcc<br>0, 3.5V(Default)<br>1, 2.5V<br>2, 1.8V<br>1800 ~ 3800, 1.8 ~ 3.8V (minimum step 100mV)<br>(For SF600/SF600*Plus* only) |

| | |
|---|---|
| --vpp | - work with --prog and --erase.<br>apply vpp when the memory chip supports it |
| --log | write operation result into file<br>"%appdata%\dediprog\SF100\log.txt" |
| -i [ --silent ] | suppress the display of real-time timer counting<br>- used when integrating with 3rd-party tools (e.g. IDE) |
| --spi-clk arg (=2) | specify SPI clock:<br>2, 12 MHz(Default)<br>0, 24 MHz<br>1, 8 MHz<br>3, 3 MHz<br>4, 2.18 MHz<br>5, 1.5 MHz<br>6, 750 KHz<br>7, 375 KHz |

## 5.6 Exit Code

```
enum ErrorCode
{
EXCODE_PASS,
EXCODE_FAIL_ERASE,
EXCODE_FAIL_PROG,
EXCODE_FAIL_VERIFY,
EXCODE_FAIL_READ,
EXCODE_FAIL_BLANK,  // 5
EXCODE_FAIL_BATCH,
EXCODE_FAIL_CHKSUM,
EXCODE_FAIL_IDENTIFY,
EXCODE_FAIL_OTHERS=99,
};
```

# VI. Specific Functions (SF600 and SF600*Plus*)

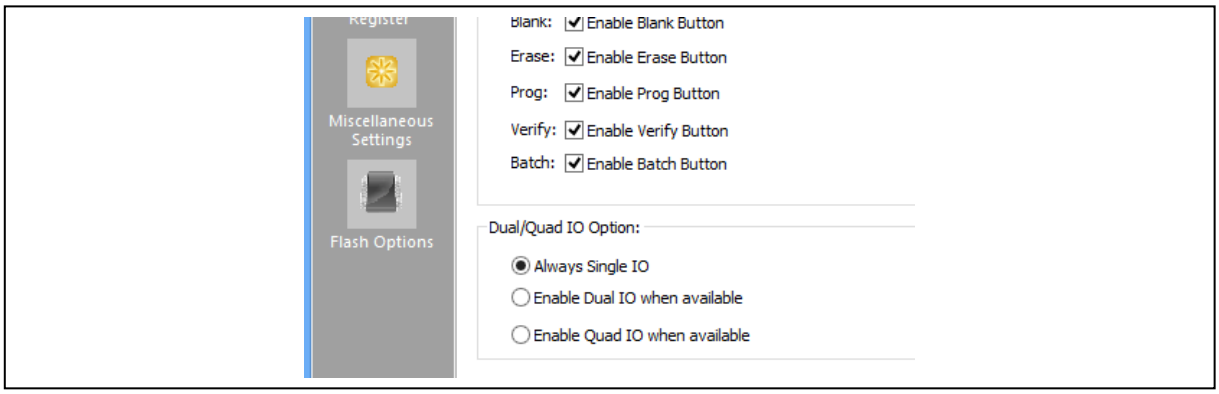
## 6.1Dual/Quad IO

User can find Dual/Quad IO option in **Config/Miscellaneous Settings**.

The default of Dual/Quad IO option is “Always Single IO”.

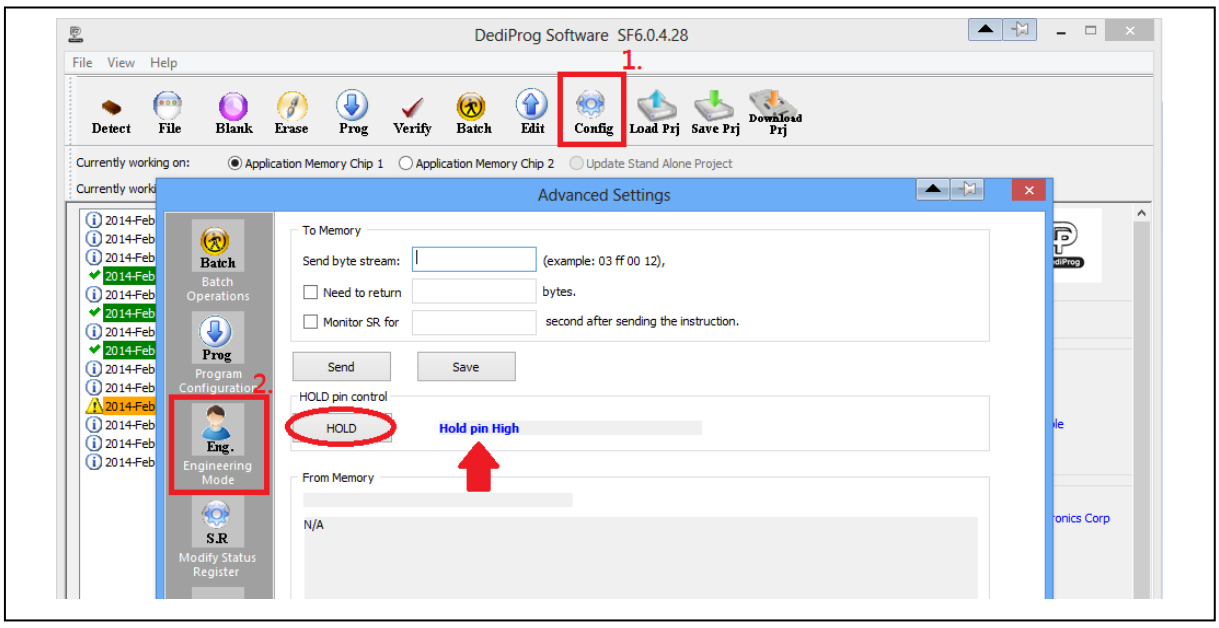
SF600 and SF600*Plus* support Dual and Quad IO. When using a SPI Flash with Dual/Quad IO function, user can select Dual or Quad IO mode.

**Note:** Socket mode only support Single/Dual IO mode. When use the socket programming the Quad IO function will disable.



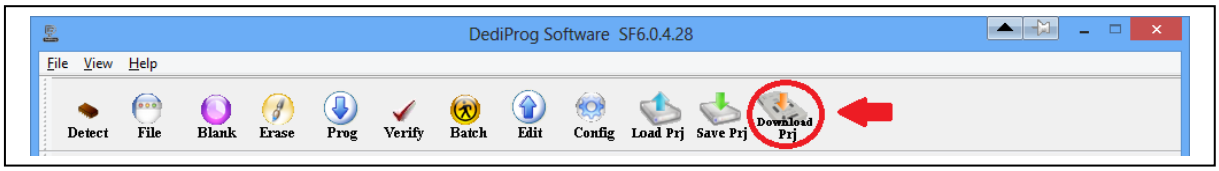

## 6.2 Hold Pin Status Setting

SF600 and SF600*Plus* programmer is available to set HOLD pin status through software. Please go through Config and change it under engineering mode. Click on HOLD button to change the status of HIGH or LOW. This function is available at SF600 and SF600*Plus.*



# VII. Stand Alone Mode (SF600*Plus* only)

In addition to the functions provided by SF100 and SF600, SF600*Plus* further allow users download project to SF600*Plus* directly and to program serial flash memories in the standalone mode.

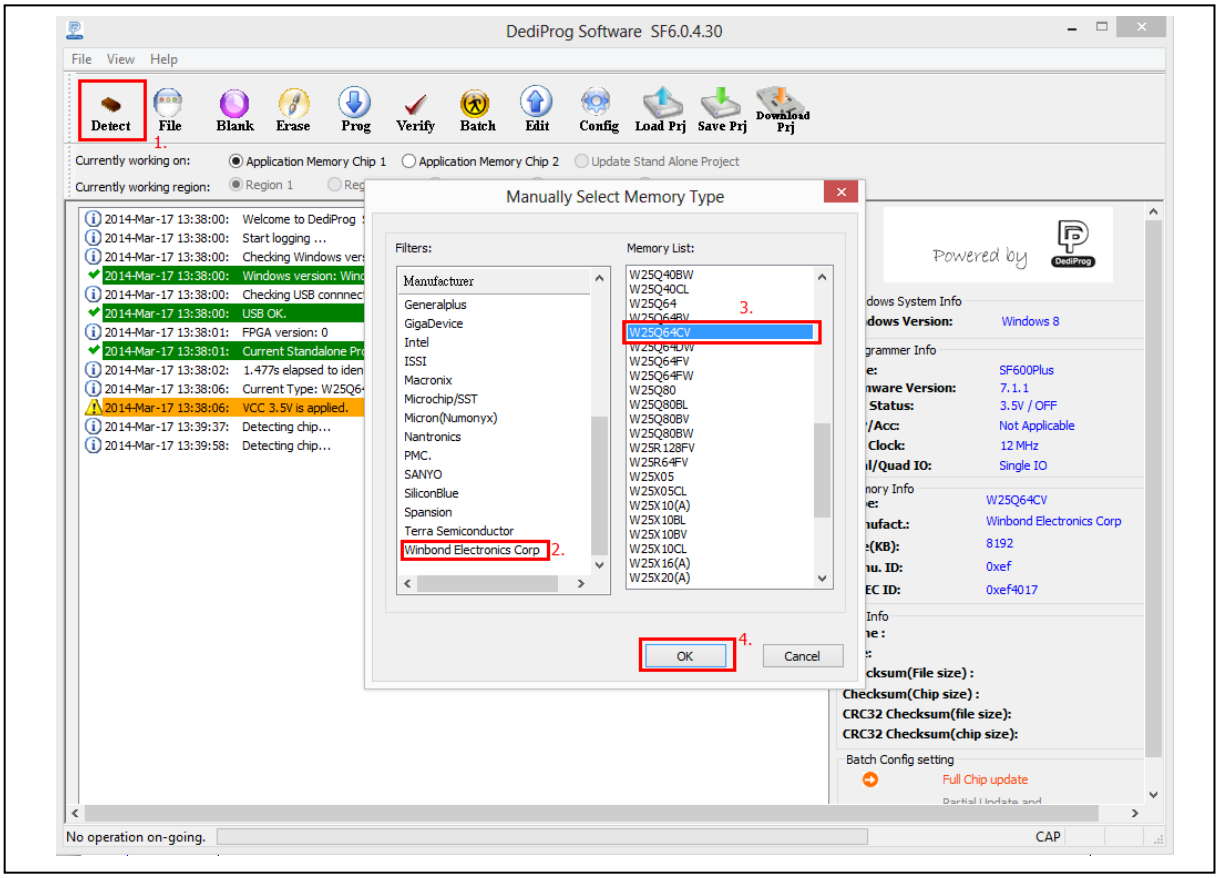

DediProg Software SF6.0.4.28
File View Help
Detect File Blank Erase Prog Verify Batch Edit Config Load Prj Save Prj Download Prj

## 7.1 Project preparation

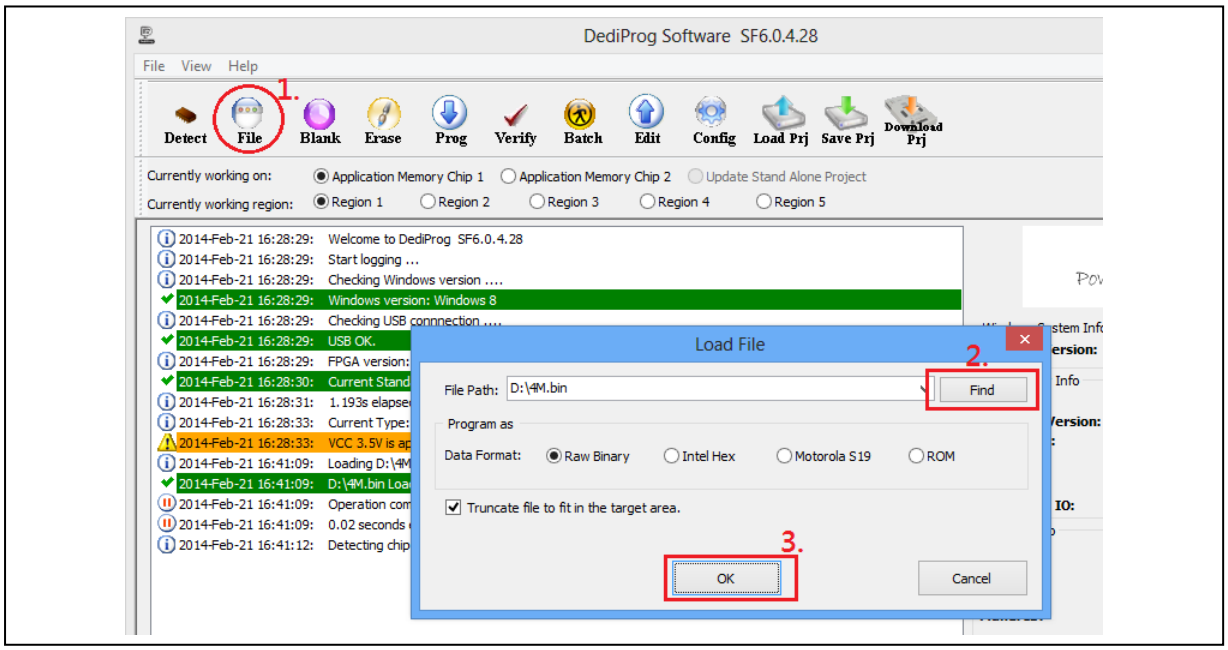
Prepare a stand alone programming project.

7.1.1 Open DediProg Engineer software.

7.1.2 Select IC brand and part number.



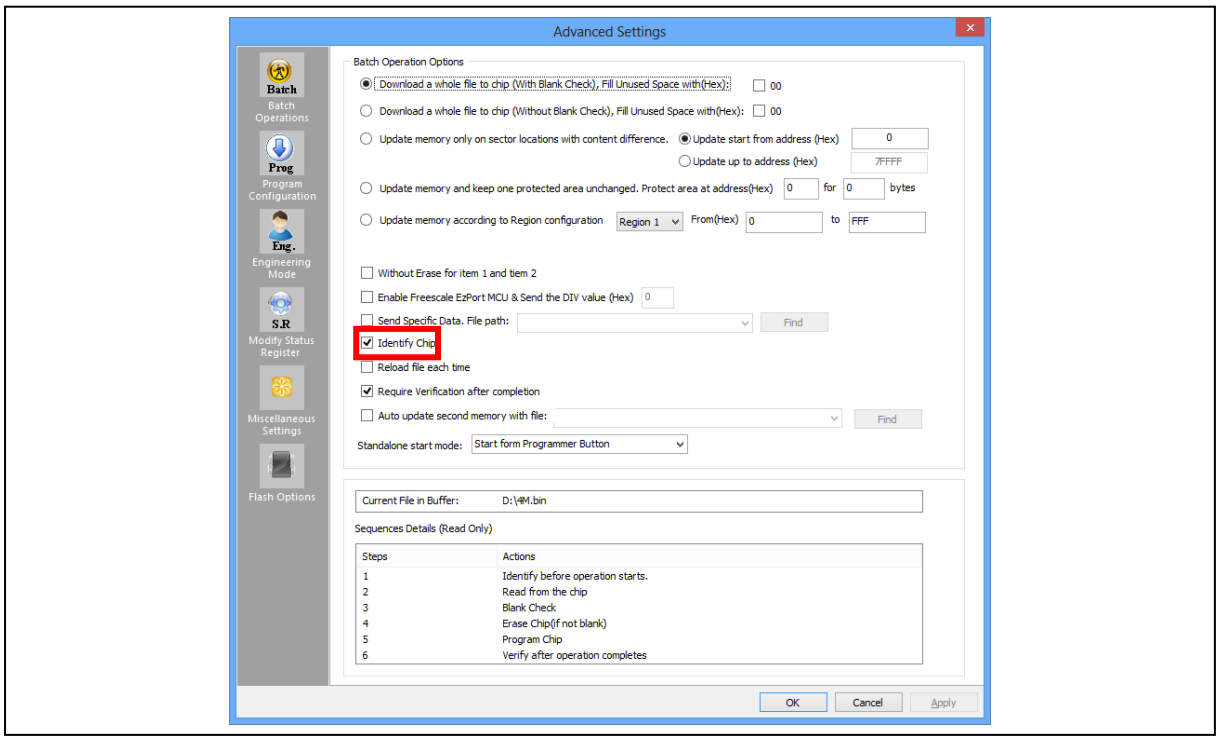
7.1.3 Load the programing file.


DediProg Software SF6.0.4.28
File View Help
1.
Detect File Blank Erase Prog Verify Batch Edit Config Load Prj Save Prj Download Prj
Currently working on: Application Memory Chip 1 Application Memory Chip 2 Update Stand Alone Project
Currently working region: Region 1 Region 2 Region 3 Region 4 Region 5
2014-Feb-21 16:28:29: Welcome to DediProg SF6.0.4.28
2014-Feb-21 16:28:29: Start logging ...
2014-Feb-21 16:28:29: Checking Windows version ....
2014-Feb-21 16:28:29: Windows version: Windows 8
2014-Feb-21 16:28:29: Checking USB connnection ....
2014-Feb-21 16:28:29: USB OK.
2014-Feb-21 16:28:29: FPGA version:
2014-Feb-21 16:28:30: Current Stand
2014-Feb-21 16:28:31: 1.193s elapse
2014-Feb-21 16:28:33: Current Type:
2014-Feb-21 16:28:33: VCC 3.5V is ap
2014-Feb-21 16:41:09: Loading D:\4M
2014-Feb-21 16:41:09: D:\4M.bin Loa
2014-Feb-21 16:41:09: Operation com
2014-Feb-21 16:41:09: 0.02 seconds
2014-Feb-21 16:41:12: Detecting chip
Load File
2.
File Path: D:\4M.bin
Find
Program as
Data Format: Raw Binary Intel Hex Motorola S19 ROM
Truncate file to fit in the target area.
3.
OK
Cancel

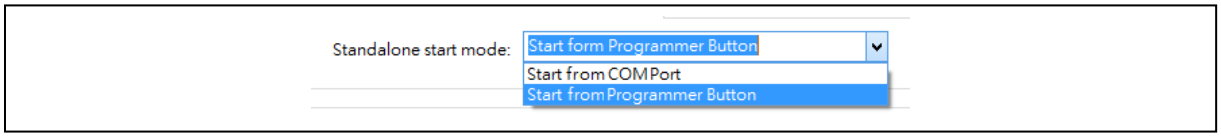
7.1.4 Click “Config” Icon to set programming flow.

**Important Notice:**

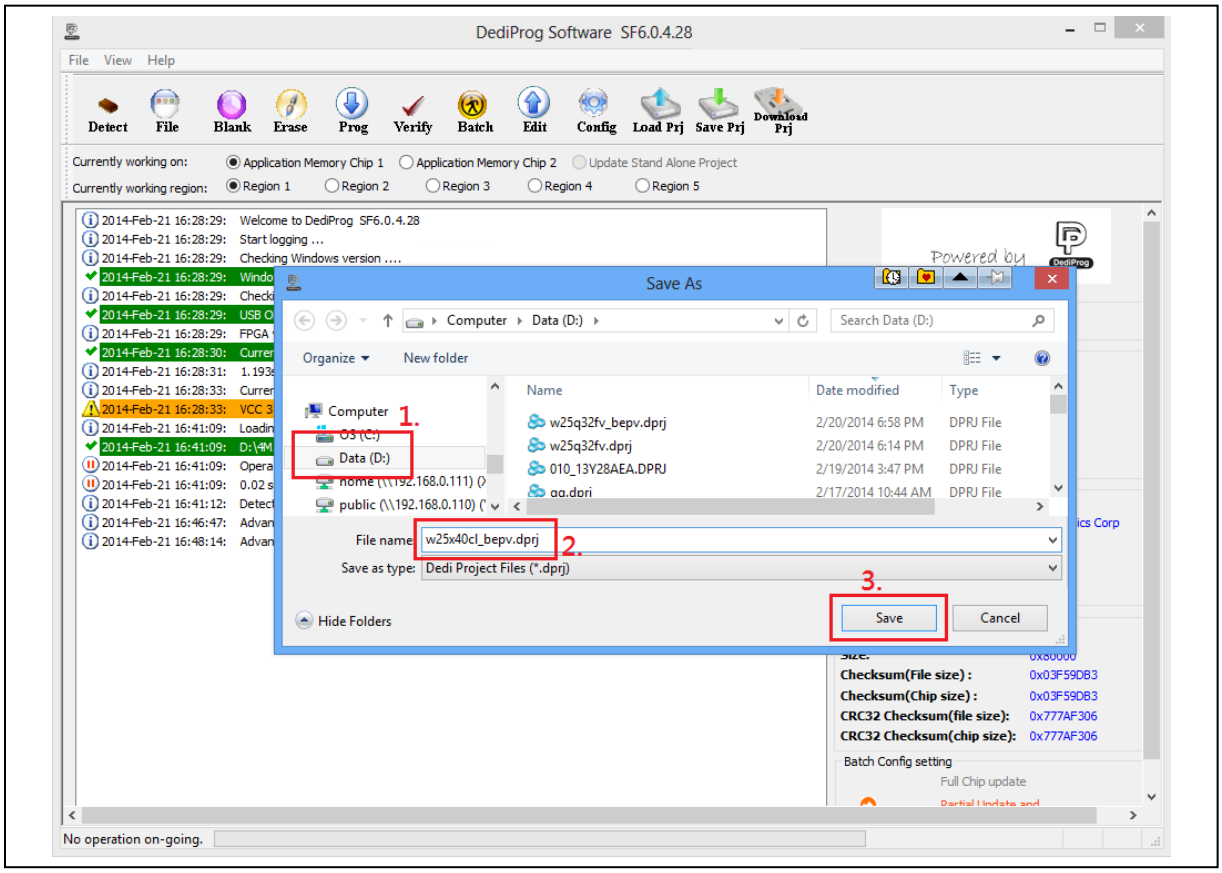
**“Identify Chip” is necessary for SF600*plus* stand alone programming.**
**Be sure to include “Identify Chip” in programming flow.**


Advanced Settings
Batch
Batch Operations
Prog
Program Configuration
Eng.
Engineering Mode
S.R
Modify Status Register
Miscellaneous Settings
Flash Options
Batch Operation Options
Download a whole file to chip (With Blank Check), Fill Unused Space with(Hex): 00
Download a whole file to chip (Without Blank Check), Fill Unused Space with(Hex): 00
Update memory only on sector locations with content difference.
Update start from address (Hex) 0
Update up to address (Hex) 7FFFF
Update memory and keep one protected area unchanged. Protect area at address(Hex) 0 for 0 bytes
Update memory according to Region configuration Region 1 From(Hex) 0 to FFF
Without Erase for item 1 and tiem 2
Enable Freescale EzPort MCU & Send the DIV value (Hex) 0
Send Specific Data. File path:
Find
Identify Chip
Reload file each time
Require Verification after completion
Auto update second memory with file:
Find
Standalone start mode: Start form Programmer Button
Current File in Buffer: D:\4M.bin
Sequences Details (Read Only)
Steps Actions
1 Identify before operation starts.
2 Read from the chip
3 Blank Check
4 Erase Chip(if not blank)
5 Program Chip
6 Verify after operation completes
OK
Cancel
Apply

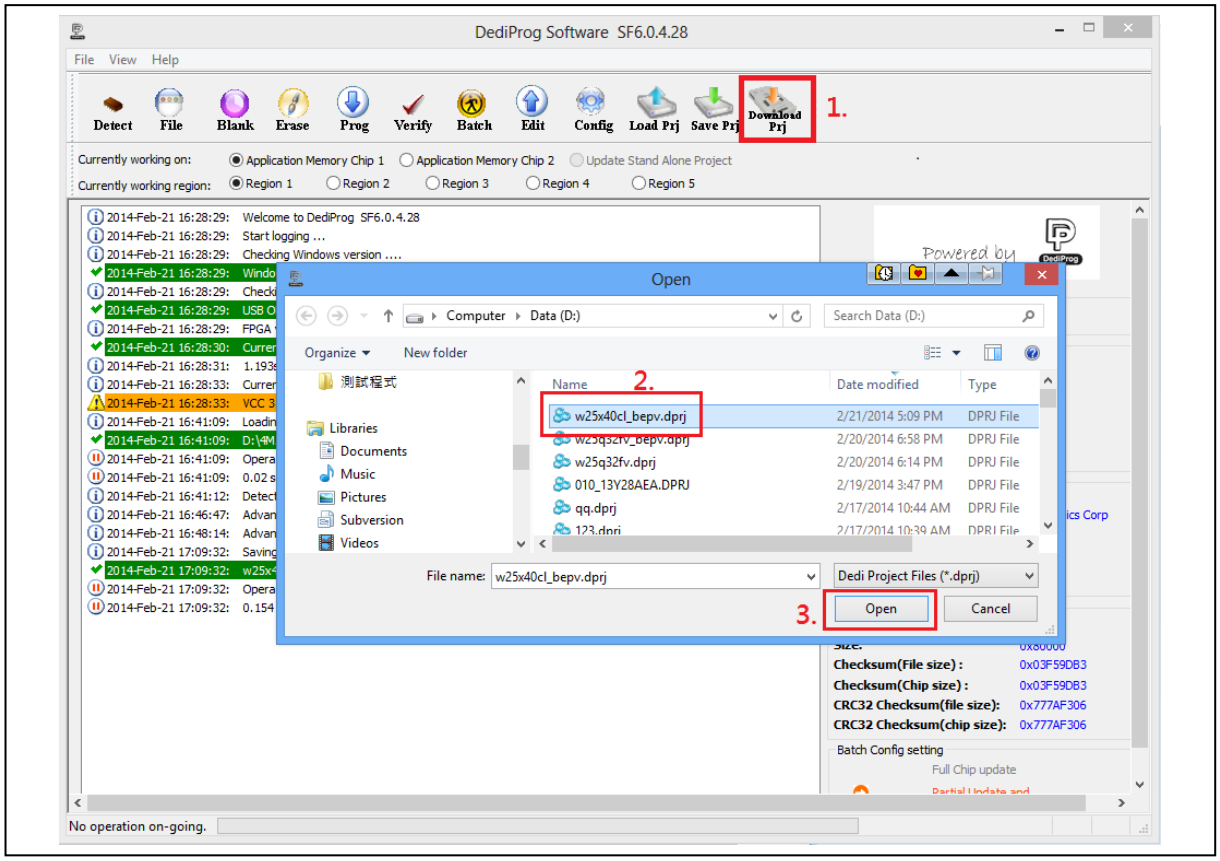
### 7.1.5 Choosing Stand Alone start mode



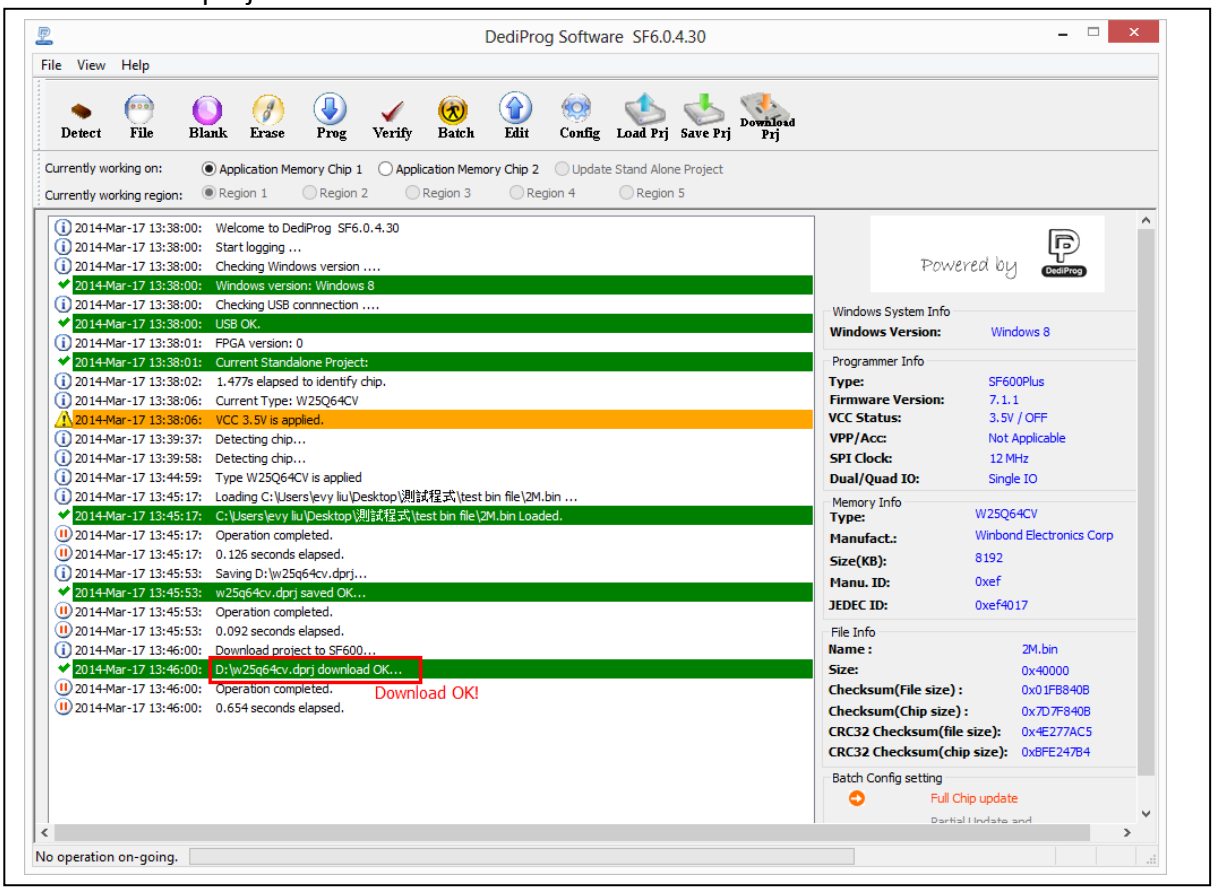

### 7.1.6 Save dprj file to PC.


DediProg Software SF6.0.4.28
File View Help
Detect File Blank Erase Prog Verify Batch Edit Config Load Prj Save Prj Download Prj
Currently working on: Application Memory Chip 1 Application Memory Chip 2 Update Stand Alone Project
Currently working region: Region 1 Region 2 Region 3 Region 4 Region 5
2014-Feb-21 16:28:29: Welcome to DediProg SF6.0.4.28
2014-Feb-21 16:28:29: Start logging ...
2014-Feb-21 16:28:29: Checking Windows version ....
Powered by
Save As
Computer Data (D:)
Search Data (D:)
Organize New folder
Computer
Data (D:)
Name
Date modified
Type
w25q32fv_bepv.dprj 2/20/2014 6:58 PM DPRJ File
w25q32fv.dprj 2/20/2014 6:14 PM DPRJ File
010_13Y28AEA.DPRJ 2/19/2014 3:47 PM DPRJ File
1.
2.
3.
File name: w25x40cl_bepv.dprj
Save as type: Dedi Project Files (*.dprj)
Hide Folders
Save
Cancel
Checksum(File size) : 0x03F59DB3
Checksum(Chip size) : 0x03F59DB3
CRC32 Checksum(file size): 0x777AF306
CRC32 Checksum(chip size): 0x777AF306
Batch Config setting
Full Chip update
No operation on-going.

7.1.7 Press "**Download Prj**" button to download project to SF600*Plus* embedded memory




7.1.8 Download project successful


DediProg Software SF6.0.4.30
File View Help
Detect File Blank Erase Prog Verify Batch Edit Config Load Prj Save Prj Download Prj
Currently working on: Application Memory Chip 1 Application Memory Chip 2 Update Stand Alone Project
Currently working region: Region 1 Region 2 Region 3 Region 4 Region 5
2014-Mar-17 13:38:00: Welcome to DediProg SF6.0.4.30
2014-Mar-17 13:38:00: Start logging ...
2014-Mar-17 13:38:00: Checking Windows version ....
2014-Mar-17 13:38:00: Windows version: Windows 8
2014-Mar-17 13:38:00: Checking USB connnection ....
2014-Mar-17 13:38:00: USB OK.
2014-Mar-17 13:38:01: FPGA version: 0
2014-Mar-17 13:38:01: Current Standalone Project:
2014-Mar-17 13:38:02: 1.477s elapsed to identify chip.
2014-Mar-17 13:38:06: Current Type: W25Q64CV
2014-Mar-17 13:38:06: VCC 3.5V is applied.
2014-Mar-17 13:39:37: Detecting chip...
2014-Mar-17 13:39:58: Detecting chip...
2014-Mar-17 13:44:59: Type W25Q64CV is applied
2014-Mar-17 13:45:17: Loading C:\Users\evy liu\Desktop\測試程式\test bin file\2M.bin ...
2014-Mar-17 13:45:17: C:\Users\evy liu\Desktop\測試程式\test bin file\2M.bin Loaded.
2014-Mar-17 13:45:17: Operation completed.
2014-Mar-17 13:45:17: 0.126 seconds elapsed.
2014-Mar-17 13:45:53: Saving D:\w25q64cv.dprj...
2014-Mar-17 13:45:53: w25q64cv.dprj saved OK...
2014-Mar-17 13:45:53: Operation completed.
2014-Mar-17 13:45:53: 0.092 seconds elapsed.
2014-Mar-17 13:46:00: Download project to SF600...
2014-Mar-17 13:46:00: D:\w25q64cv.dprj download OK...
2014-Mar-17 13:46:00: Operation completed.
2014-Mar-17 13:46:00: 0.654 seconds elapsed.
Download OK!
Powered by
Windows System Info
Windows Version: Windows 8
Programmer Info
Type: SF600Plus
Firmware Version: 7.1.1
VCC Status: 3.5V / OFF
VPP/Acc: Not Applicable
SPI Clock: 12 MHz
Dual/Quad IO: Single IO
Memory Info
Type: W25Q64CV
Manufact.: Winbond Electronics Corp
Size(KB): 8192
Manu. ID: 0xef
JEDEC ID: 0xef4017
File Info
Name : 2M.bin
Size: 0x40000
Checksum(File size) : 0x01FB840B
Checksum(Chip size) : 0x7D7F840B
CRC32 Checksum(file size): 0x4E277AC5
CRC32 Checksum(chip size): 0xBFE247B4
Batch Config setting
Full Chip update
No operation on-going.

## 7.2 Stand Alone programming

Start Stand Alone programming.

### 7.2.1 **"Start from Programmer Button"** mode

Press "**Start**" button for 2 seconds to run the project in Stand Alone mode.

### 7.2.2 **"Start from COM Port"** mode

The Com Port design is for integrating SF600/SF600*Plus* with customer's system. All programmer pin outs (except 5V and NC) are default with Low status. Once customer/system sends a High signal to trigger START and make the programmer working (i.e. BUSY becomes High status accordingly), SF600/SF600*Plus* will also feedback PASS or FAIL result with High signal after programming.

# VIII. Firmware Support for Microsoft Windows

Consider the programming stability with the software and firmware on Windows 8.1.
DediProg modify the way of communication between Windows and SF serial products.
Please note the software and firmware compatibility. Before you update the firmware, please make sure your software is the latest version.
User can download the latest version on DediProg website.
**www.dediprog.com/download**

| Model Name | Current Firmware Version | New Release Version |
|---|---|---|
| **SF100** | 5.x.x | 5.5.01 |
| **SF600** | 6.x.x | 6.9.0 |
| **SF600** | 7.x.x | 7.1.0 and above version |

# IX. Revision History

| Date | Version | Changes |
|---|---|---|
| 2010/03/19 | 5.5 | Added: Enable EzPort Function on Configuration; log.txt file available on Commend line; Blink/Device/Fix-Device on Dpcmd. |
| 2010/04/14 | 5.6 | Added: Update up to address option on Batch and Program Configuration operation options. |
| 2010/05/10 | 5.7 | Minor improvement |
| 2011/05/18 | 5.8 | 1. Added specific function.<br>2. Added region configuration programming function. |
| 2011/08/26 | 5.9 | Added SF600 Hold pin status setting method. |
| 2012/01/09 | 6.0 | Added SF600 stand alone programming. |
| 2012/12/20 | 6.1 | Revise the CLI detail and add exit codes. |
| 2013/08/23 | 6.2 | 1. Added status register-2 function<br>2. Added the multiple-Dpcmd function. |
| 2013/12/18 | 6.3 | 1. Remove part of SF200 and SF300<br>2. Remove “isolation free” from software |
| 2014/02/25 | 6.4 | 1. New feature for SF600*Plus* |
| 2014/04/28 | 6.5 | 1. Replenish COM Port feature of Stand Alone mode |

**DediProg Technology Co., Ltd (Taiwan)**

4F., No.7, Ln. 143, Xinming Rd., Neihu Dist., Taipei City 114, Taiwan **TEL: 886-2-2790-7932** FAX: 886-2-2790-7916

**DediProg Technology (ShangHai)**

Room 503, Block E, No.1618, Yishan Road, Shanghai, China **TEL: 86-21-5160-0157** FAX: 86-21-6126-3530

Technical Support：**support@dediprog.com**

Sales Support：**sales@dediprog.com**

www.dediprog.com

Information furnished is believed to be accurate and reliable. However, DediProg assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties which may result from its use. Specifications mentioned in this publication are subject to change without notice.

This publication supersedes and replaces all information previously supplied.

All rights reserved Printed in Taiwan.